



**Danilo Alexandre
Estima Saraiva**

**Serviços de distribuição, recolha de dados e
indicação de presença para a bicicleta Bewegen**



**Danilo Alexandre
Estima Saraiva**

**Serviços de distribuição, recolha de dados e
indicação de presença para a bicicleta Bewegen**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Alexandre Manuel Moutela Nunes da Mota, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Mestre Rui Miguel de Bernardes Rebelo, Director do Departamento de Hardware da empresa Micro I/O – Serviços de Electrónica, Lda.

Dedico este trabalho a todos os que nunca desistiram de mim.

o júri / the jury

presidente / president

Prof. Doutor Rui Manuel Escadas Ramos Martins

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor António João Nunes Serrador

Professor Adjunto do Instituto Superior de Engenharia de Lisboa

Mestre Rui Miguel de Bernardes Rebelo

Diretor do Departamento de Hardware da empresa Micro I/O – Serviços de Electrónica, Lda

agradecimentos / acknowledgements

Ao meu orientador, Professor Doutor Alexandre Mota, pela orientação que me foi dando, por estar sempre disponível para me auxiliar, por nunca deixar que me desleixasse e por não desistir do meu trabalho.

Ao meu co-orientador, Mestre Rui Rebelo, pelas excelentes orientações técnicas que me deu, por me ter apoiado sempre no desenvolvimento deste projeto e por estar disponível para me auxiliar nos obstáculos encontrados.

À Micro I/O, nas pessoas da sua administração, por me ter tornado possível desenvolver este projeto, pondo à disposição todos os recursos da empresa, não impondo obstáculos ao desenvolvimento deste trabalho. Um especial agradecimento ao professor José Alberto Fonseca pelo tempo disponibilizado.

À minha mãe e irmão, por terem sido compreensivos pela falta de tempo para lhes dedicar, por estarem muito tempo sem a minha presença, por todas as coisas que não pude fazer por estar a trabalhar na tese.

À Adriana, amiga de longa data, por toda a amizade, por estar sempre presente, por me ter apoiado sempre que precisei, por me ouvir sempre que precisei de aliviar a pressão e por não ter desistido de mim, mesmo quando precisei de abdicar de muito tempo para poder trabalhar para esta dissertação.

À Marlene, por toda a amizade e compreensão, por nunca ter desistido de mim, mesmo sabendo que a falta de tempo pode levar a não termos tempo para tudo.

À Mariana, por toda a amizade e apoio proporcionado, por todos os momentos de desabafo e alívio quando o tempo apertava e sempre que surgiam problemas inesperados.

Aos colegas da Micro I/O, pela amizade e contributo com os seus conhecimentos, sempre que tinha a necessidade de desbloquear o pensamento.

A todas as pessoas que me incentivaram e motivaram e que não detalei nestes agradecimentos.

Obrigado, a todos vós, sem vocês este trabalho não seria possível.

Palavras Chave

Bicicletas elétricas, Bike Sharing, Bluetooth Low Energy, Android, aplicação móvel, manutenção.

Resumo

Nos últimos anos os países desenvolvidos da Europa e do Mundo têm apostado em políticas de redução da poluição ambiental e na implementação de programas que potenciam o uso das energias renováveis e a tentativa de redução do recurso aos automóveis individuais dentro das cidades, aconselhando os cidadãos a recorrer a transportes públicos ou a alternativas menos poluentes. Ao abrigo destas políticas, foram surgindo programas de bicicletas partilhadas por todo o mundo, destinados a proporcionar aos cidadãos uma alternativa mais rápida, económica e não poluente, ao mesmo tempo que obriga à realização de algum exercício físico, potenciando um bem-estar e uma melhoria da saúde. O projeto Bikeemotion surgiu como uma colaboração multidisciplinar para desenvolver um produto que potenciasse o uso da bicicleta, ao mesmo tempo que pudesse ser uma alternativa viável aos veículos automóveis.

Atualmente, a maior limitação que ocorre nestes sistemas de bicicletas partilhadas corresponde à necessidade de existir uma ligação de dados para se poder efetuar um diagnóstico das mesmas, o que obriga a uma desmontagem parcial para resolver qualquer problema existente. Dessa forma, esta dissertação aposta no desenvolvimento de um mecanismo de comunicação adicional, sobre uma ligação Bluetooth Low Energy, capaz de executar diagnósticos ao estado das bicicletas de forma mais fácil e de permitir o desenvolvimento de novas funcionalidades para o utilizador.

Keywords

e-bike, bike sharing, Bluetooth Low Energy, Android, mobile application, maintenance.

Abstract

In the last years, european countries and other developed countries in North America and Asia have focused on policies to reduce environmental pollution and started to implement programs to promote the use of renewable energies and to try to reduce the use of individual cars within cities. These last initiatives were based on advising citizens to use public transportation or trying to induce them to choose less polluting alternatives. Also, restrictions to the circulation of cars in the city centers were made harder as well as the associated costs. Under these policies, bicycling programs have been developed throughout the world to provide citizens with a faster, more economical and non-polluting alternative, while, at the same time, promoting some physical exercise, thus enhancing well-being and health conditions.

One of the projects related with this transportation was Bikeemotion which emerged as a multidisciplinary collaboration to develop a product to promote the shared usage of bicycles, in particular electric ones (e-bikes), as a viable alternative to motorized vehicles.

Currently, one of the main limitations that affects these public bicycle systems corresponds to the need for a data connection required to make the diagnosis of the bicycle hardware in the presence of defective operation. In such cases a partial disassembly is required to identify and, after, to solve any problem in them. With this idea in mind, this thesis focuses on the development of an additional communication mechanism, based on a Bluetooth Low Energy connection, able to trigger diagnosis tests to verify the operational state of the bicycles, in an easier way, connecting to a mobile phone app used by the technician. This BLE interface allows also the development of new functionalities for the user, such as the off-line lock and unlock of the bike.

Conteúdo

Conteúdo	i
Lista de Figuras	v
Lista de Tabelas	vii
Glossário	ix
1 Introdução	1
1.1 Bicicleta Elétricas	1
1.2 Motivação	2
1.3 Objetivos	3
1.4 Solução Proposta	4
1.5 Estrutura	4
2 Estado da Arte	5
2.1 Sistemas de Bicicletas Partilhadas	5
2.1.1 Disponibilização das bicicletas ao utilizador	7
2.1.2 Soluções Comerciais	8
2.1.3 Bikeemotion	10
2.1.4 BIXI Montréal	13
2.1.5 BiciMad	14
2.1.6 Bycyklen	15
2.1.7 ESB	17
2.1.8 Velobility	18
2.1.9 Lidl Bike	19
2.1.10 Análise Comparativa das soluções	20
2.2 Redes de Área Pessoal	22
2.2.1 ZigBee	23
2.2.2 Ligação por Infravermelhos	24

2.2.3	Bluetooth Classic	25
2.2.4	Bluetooth Low Energy	29
3	Arquitetura do Sistema Inicial	41
3.1	Origem e evolução do projeto	41
3.2	Hardware	42
3.2.1	Placa-mãe	43
3.2.2	Placa de comunicações	45
3.3	Plataformas de gestão	46
3.4	Bicicleta	46
3.5	Modelo de funcionamento	47
3.6	Avaliação do sistema	47
4	Trabalho desenvolvido	49
4.1	Requisitos do desenvolvimento	49
4.2	Prova de Conceito	50
4.2.1	Hardware	50
4.2.2	Firmware	51
4.2.3	Aplicação Android	52
4.3	Hardware	53
4.4	Firmware	58
4.5	Gestão de Alimentação	59
5	Aplicação Móvel Android	61
5.1	Sistema Operativo Android	61
5.1.1	Detalhes da Plataforma Android	63
5.2	Atividades	65
5.2.1	Login	65
5.2.2	Mapa em tempo real	66
5.2.3	Estações	66
5.2.4	Veículos	67
5.2.5	Quiosques	67
5.2.6	Eventos	68
5.2.7	Rotas	68
5.2.8	Bluetooth	69
5.3	Permissões	70
5.4	Serviço de Bluetooth Low Energy	71
5.5	Gestor de Mensagens	72
5.6	Processador de Comandos	73

5.7	Scanner Bluetooth Low Energy	73
5.8	Broadcast Receiver	74
6	Protocolo de Comunicação	75
6.1	Estrutura de comandos	75
6.2	Comandos	76
6.3	Subsistema Bike	76
6.3.1	Tampering	77
6.3.2	Acelerómetro	78
6.3.3	Temperatura	78
6.3.4	Bateria	78
6.3.5	Odómetro	79
6.3.6	LDR	79
6.3.7	Reset	80
6.3.8	Locks	80
6.3.9	EEPROM	81
6.3.10	Tech Screen	81
6.3.11	Motor Control	81
6.3.12	USB	82
6.3.13	Torque	82
6.3.14	Assistência	83
6.3.15	Áudio	83
6.3.16	GPS	83
6.3.17	Luzes	84
6.3.18	Modem	84
6.3.19	RFID	85
6.4	Subsistema Extended	85
6.4.1	Versão	85
6.4.2	Bike Info	86
6.4.3	Message Handler	86
6.4.4	Action	86
6.4.5	Restart	87
6.4.6	Modo técnico	87
7	Testes e Validação	89
7.1	Testes de desempenho	89
7.1.1	Consumo energético	89
7.1.2	Estabilidade da ligação	90

7.1.3	Funcionamento com mensagens grandes	90
7.1.4	Alcance	91
7.1.5	Largura de Banda	92
7.2	Análise dos Resultados	93
8	Considerações Finais	95
8.1	Conclusões	95
8.2	Trabalho Futuro	95
	Referências	97

Lista de Figuras

1.1	Evolução da assistência em bicicletas Pedelec ¹ [3]	2
2.1	Bicicletas deixadas em caixotes do lixo[6]	7
2.2	Bicicletas depositadas no lago Merrit[7]	8
2.3	Evolução das bicicletas partilhadas por tipo nos EUA[9]	9
2.4	Número de bicicletas partilhadas e viagens realizadas entre 2010 e 2016[11]	9
2.5	Mapa da distribuição dos sistemas de bicicletas partilhadas entre 2010 e 2016[12]	10
2.6	Bicicleta Bewegen[15]	12
2.7	Docas e quiosque Bixi[16]	14
2.8	Interface com o utilizador das bicicletas BiciMad[17]	15
2.9	Interface tátil com o utilizador das bicicletas Bycyklen[18]	16
2.10	Aplicação Móvel ESB[19]	18
2.11	Encapsulamento dos componentes mecânicos e eletrónicos[20]	19
2.12	Zona de parque do sistema Lidl Bike[29]	20
2.13	Arquitetura Bluetooth 3.0[32]	26
2.14	Exemplo de uma piconet com 1 escravo, multi-escravo e scatternet[32]	27
2.15	Comunicação entre mestre e escravo[32]	28
2.16	Arquitetura Bluetooth 4.0[32]	28
2.17	Interação entre sistemas Bluetooth[33]	29
2.18	Pilha Protocolar BLE ² [32]	30
2.19	Canais de comunicação BLE[34]	31
2.20	Máquina de Estados BLE[35]	32
2.21	Sequência de ligação mestre-escravo BLE[36]	37
2.22	Formato geral de um pacote BLE[32]	37
2.23	Formato do PDU no canal de anúncio[32]	38
2.24	Formato do Header do PDU no canal de anúncio[32]	38
2.25	Formato do PDU no canal de dados[32]	39

¹Pedal Electric Cycle

²Bluetooth Low Energy

2.26	Formato do Header do PDU no canal de dados[32]	39
2.27	Formato do <i>payload</i> do PDU de dados[32]	40
2.28	Formato do cabeçalho do pacote L2CAP[32]	40
3.1	Sistema inicial Bikeemotion	42
3.2	Diagrama de blocos do projeto Bikeemotion	43
3.3	Placa-mãe	44
3.4	Placa de comunicações	45
3.5	Bicicleta Bewegen[15]	47
4.1	Implementação do módulo na prova de conceitos	51
4.2	Ecrã de pesquisa de dispositivos da aplicação de teste	52
4.3	Ecrã de comandos da aplicação de teste	53
4.4	Referência para desenho de esquemático do módulo RN4020	54
4.5	Esquemático da implementação BLE	54
4.6	Referência para desenho de layout do módulo RN4020	55
4.7	Placa de Circuito Impresso com a implementação do módulo RN4020	55
4.8	Esquemático do circuito do controlo de alimentação do módulo BLE	57
4.9	Máquina de Estados da aplicação	59
5.1	Arquitetura do Android[43]	62
5.2	Ciclo de vida de uma atividade[46]	64
5.3	Atividade de login	65
5.4	Atividade do mapa de tempo real	66
5.5	Atividade de lista de estações	66
5.6	Atividade de lista de veículos	67
5.7	Atividade de lista de quiosques	67
5.8	Atividade de lista de eventos	68
5.9	Atividade de lista de rotas	68
5.10	Atividade de pesquisa de dispositivos Bluetooth Low Energy	69
5.11	Atividade de envio de comandos para as bicicletas	70
5.12	Janela de permissões explícitas do sistema Android	71
7.1	RSSI em função da distância	92
7.2	Largura de banda do sistema	93

Lista de Tabelas

2.1	Sistemas atualmente implementados pela Bikeemotion	11
2.2	Tabela das funcionalidades disponibilizadas por cada sistema	21
2.3	Exemplo de definição de dois serviços	35
2.4	Exemplo de definição de duas características para um serviço	35
2.5	Incorporação do descritor CCC no exemplo do serviço 1	36
2.6	Subtipos de PDU do canal de anúncios	39
4.1	Ligações entre portos do módulo RN4020 e do microcontrolador	56
5.1	Eventos de notificação do serviço BLE	72
6.1	Tabela dos processadores de comandos, respetivo ID na trama e descrição da sua função	76
6.2	Tabela dos comandos existentes no subsistema Bike, respetivo ID na trama e descrição da sua função	77
6.3	Tabela dos comandos existentes no subsistema extended, respetivo ID na trama e descrição da sua função	85
7.1	Consumo do sistema com o estado da ligação BLE	90
7.2	Dados obtidos do RSSI para cada valor de distância	91

Glossário

BUGA	Bicicleta de Utilização Gratuita de Aveiro
BLE	Bluetooth Low Energy
GATT	Generic Attribute
UART	Universal Asynchronous Receiver/Transmitter
MLDP	Microchip Low-energy Data Protocol
Pedelec	Pedal Electric Cycle
GPS	Global Position System
IDT	Investigação e Desenvolvimento Tecnológico
PAN	Personal Area Network
IrDA	Infrared Data Association
ISM	Industrial, Scientific and Medical
SIG	Special Interest Group
FHSS	Frequency-Hopping Spread Spectrum
GFSK	Gaussian Frequency-Shift Keying
DQPSK	Differential Quadrature Phase-Shift Keying
DPSK	Differential Phase-Shift Keying
EDR	Enhanced Data Rate
BR	Basic Rate
IEEE	Institute of Electrical and Electronics Engineers
RFID	Identificação por Rádio-Frequência
WPAN	Wireless Personal Area Network
3G	3ª Geração de Redes Móveis
LCD	Liquid Crystal Display
LED	Light Emitting Diode
TCP	Transport Control Protocol
TICE	Tecnologias de Informação, Comunicação e Eletrónica
RAM	Random Access Memory
ASCII	American Standard Code for Information Interchange
NFC	Near Field Communication
DKK	Coroa Dinamarquesa
B2B	Business to Business
GPRS	General Packet Radio Service
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuits
NACTO	National Association of City Transportation Officials

EUA	Estados Unidos da América
API	Aplication Programming Interface
IDC	International Data Corporation
WMAN	Wireless Metropolitan Area Network
WWAN	Wireless Wide Area Network
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
AMP	Alternate PHY/MAC
HS	High Speed
PAL	Protocol Adaptation Layer
HCI	Host Controller Interface
MAC	Media Access Control
ULP	Ultra Low Power
L2CAP	Logical Link Control and Adaptation Protocol
USB	Universal Serial Bus
ATT	Attribute Protocol
AES	Advanced Encryption Standard
UUID	Universally Unique Identifier
CCC	Client Characteristic Configuration
LE	Low Energy
PDU	Protocol Data Unit
GAP	Generic Access Profile
RFU	Reserved for Future Use
MIC	Message Integrity Check
CRC	Cyclic Redundancy Check
TDM	Time Division Multiplexing
RF	Radio-Frequência
BER	Bit Error Rate
FSK	Frequency Shift Keying
BE	Bikeemotion
SDK	Software Development Kit
IPC	Inter-Process Communication
RTS	Ready To Send
CTS	Clear To Send
SMA	SubMiniature version A
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
RSSI	Received Signal Strength Indication
PIN	Personal Identification Number
JSON	JavaScript Object Notation
SD	Secure Digital
VM	Virtual Machine
ART	Android Runtime

Introdução

1.1 BICICLETA ELÉTRICAS

As bicicletas elétricas correspondem a um meio de transporte em que a bicicleta é equipada com um sistema de baterias e um motor elétrico que lhe permitem mover-se sem a interação do utilizador ou fornecendo-lhe assistência durante a sua pedalada.

As bicicletas *Pedelec*, também denominadas de bicicletas de assistência ao pedalar, são um tipo de bicicletas elétricas que necessita que o seu utilizador pedale para que exista assistência. Desenvolvidas devido à regulamentação existente nalguns países, como os pertencentes à União Europeia, neste tipo de bicicletas, à semelhança de uma bicicleta convencional, o movimento da bicicleta é originado recorrendo à força mecânica do seu utilizador, havendo uma assistência ao mesmo recorrendo a um motor elétrico que auxilia o utilizador sempre que ocorre o pedalar. Ao contrário de uma bicicleta puramente elétrica, cuja propulsão é obtida recorrendo a um motor elétrico acionado por um acelerador elétrico ou mecânico colocado no guiador da bicicleta, no caso das bicicletas *Pedelec*, esta propulsão apenas ocorre quando o utilizador pedala na bicicleta, deixando de atuar quando este pressiona o travão ou deixa de pedalar. Este tipo de bicicletas é habitualmente dotado de sensores de torque que lhes permitem avaliar a condição de esforço do seu utilizador, aumentando a intensidade da assistência sempre que seja necessário. Derivado do fato de que a ação dos motores apenas é exercida quando existe uma interação do utilizador, semelhante a uma bicicleta tradicional, as bicicletas *Pedelec* são habitualmente tratadas como sendo legalmente equivalente a bicicletas tradicionais, na maioria dos países, enquanto que as bicicletas elétricas com acelerador são consideradas veículos motorizados, sempre que a potência dos motores excede os 250W.[1]

O sistema de assistência das bicicletas *Pedelec* dá origem a uma maior proteção das baterias, aliado a um menor consumo das mesmas, quando comparado com as bicicletas elétricas, permitindo uma maior autonomia e viagens mais longas. De forma geral, uma bicicleta *Pedelec* tem um comportamento semelhante a uma bicicleta com pedais. O motor existente serve para aumentar a potência fornecida pelo utilizador à bicicleta, e não para a substituir.

Estas bicicletas normalmente permitem ajustar o nível de assistência que é possível receber à medida que se pedala, sendo que a assistência pode ser definida entre o nível 0 e o nível máximo suportado pelo fabricante, que normalmente disponibiliza 3 ou 5 níveis, apesar deste valor poder variar de sistema para sistema. Este comportamento é ilustrado na figura 1.1, em que é possível observar que o contributo da potência dos motores elétricos das bicicletas para potência global do sistema vai diminuindo à medida que a velocidade aumenta. Ao mesmo tempo, é também possível observar que, após uma certa velocidade, o motor elétrico é automaticamente desligado, correspondendo a velocidade a que se dá este corte ao valor de 25 km/h nos países da União Europeia, como regulado pela diretiva comunitária 2002/24/CE.[2]

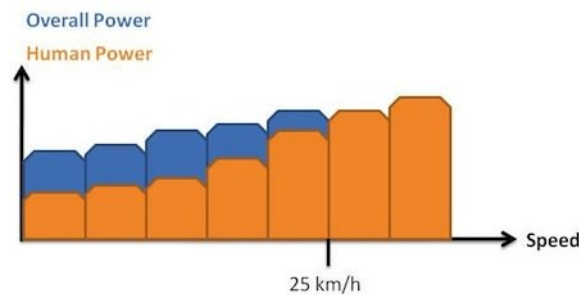


Figura 1.1: Evolução da assistência em bicicletas Pedelec[3]

1.2 MOTIVAÇÃO

Nos últimos anos, os países desenvolvidos têm apostado em políticas de redução da poluição ambiental e na implementação de programas que potenciam o uso das energias renováveis e na tentativa de redução do recurso aos automóveis individuais dentro das cidades, aconselhando os cidadãos a recorrer a transportes públicos ou a alternativas menos poluentes. Ao abrigo destas políticas, foram surgindo programas de bicicletas partilhadas por todo o mundo, destinados a proporcionar aos cidadãos uma alternativa de utilização mais rápida, económica e não poluente, ao mesmo tempo que obriga à realização de algum exercício físico, potenciando bem-estar e uma melhoria da saúde. Uma das iniciativas de relevo, na cidade de Aveiro, é o projeto BUGA¹, que permite aos seus utilizadores a recolha de uma bicicleta, num ponto central, e a sua devolução num dos diversos pontos espalhados pela cidade.

Devido aos diversos relevos que existem nas cidades, os projetos de bicicletas partilhadas mais simples, apenas com bicicletas tradicionais, têm mais dificuldade em proliferar quando a inclinação do terreno da cidade em que seriam implementados é acentuada. Por forma a solucionar este problema, foram surgindo soluções que empregam bicicletas elétricas, de assistência ao pedalar ou totalmente elétricas, que retiram do utilizador o esforço acrescido com o aumento da inclinação do pavimento.

É neste sentido que surgiu, por parte da Micro I/O, enquanto parte integrante do consórcio Bikeemotion, um projeto para o desenvolvimento da parte eletrónica para bicicletas elétricas de assistência ao pedalar, que permite resolver este problema, oferecendo ajuda ao utilizador

¹Bicicleta de Utilização Gratuita de Aveiro

nos terrenos mais íngremes, proporcionando, em todo o tipo de inclinações, uma experiência semelhante ao de uma bicicleta tradicional num terreno plano.

Esta dissertação surge como parte integrante da constante necessidade de atualização e melhoria das funcionalidades existentes e da análise e implementação de novas funcionalidades que ajudam a colmatar algumas das necessidades que são transmitidas pelas equipas de desenvolvimento, da área comercial e de manutenção do sistema Bikeemotion. O sistema, composto por toda a eletrónica, bicicleta, quiosques e docas de carregamento, assim como pela plataforma de gestão, será doravante denominado por sistema Bikeemotion, ainda que toda a componente da bicicleta e motor elétrico sejam fornecidos pela Bewegen Technologies Inc.

O sistema Bikeemotion apresenta-se como uma solução completa para planeamento, implementação e gestão de frotas de bicicletas tradicionais e elétricas de assistência. Deste modo, todas as unidades enviadas para o terreno são equipadas com um conjunto de eletrónica de controlo e monitorização. Esta eletrónica, para cumprir com os padrões de qualidade, tem que ser testada elétrica e funcionalmente antes de ser colocada efetivamente em produção numa bicicleta.

Atualmente, os testes elétricos são automatizados, mas os testes funcionais são realizados manualmente e de forma individual, o que condiciona a velocidade a que é possível fabricar a eletrónica, desde a impressão das placas de circuito impresso até ao ponto em que estas podem ser colocadas em circulação nas bicicletas.

Por forma a conseguir minimizar esta limitação, é necessário ter disponível uma plataforma de testes que permita realizar todo o diagnóstico de entradas e saídas digitais e analógicas da eletrónica, ao mesmo tempo que não exija que o teste seja realizado numa bicicleta completamente montada.

Além desta questão, durante a manutenção das bicicletas no terreno, os técnicos que realizam a manutenção dos equipamentos no terreno, não têm à sua disposição um mecanismo para a verificação do estado dos sensores e atuadores das bicicletas, sem recurso ao acesso aos dados recebidos pela plataforma de gestão do sistema, de forma ágil e simples, o que limita a sua capacidade de resolução de problemas. Este problema pode ser resolvido com o desenvolvimento de uma aplicação móvel compatível com o sistema operativo Android, recorrendo a tecnologias de comunicação de curto alcance, que permita um diagnóstico técnico centralizado, não sendo necessário recorrer à plataforma de gestão online.

1.3 OBJETIVOS

Tendo em conta a evolução e aumento do número de soluções de bicicletas partilhadas existentes em todo o mundo, o trabalho realizado ao longo desta dissertação tem como objetivo a análise das soluções existentes no mercado, permitindo a avaliação das características e funcionalidades de cada um destes sistemas, comparando-as com as existentes na solução em que este documento se baseia. Aliado a isto, pretende-se a implementação de algumas funcionalidades adicionais, que se revelam necessárias para a agilidade nas manutenções dos equipamentos, quando distribuídos pelos seus pontos de carga nas cidades que implementam a solução Bikeemotion.

Uma das inovações que serão tidas em conta consiste na implementação de um mecanismo de comunicação de curto alcance, baseado em tecnologias de comunicação sem fios Bluetooth Low Energy, que permite que seja efetuada uma comunicação entre a eletrónica das bicicletas e a aplicação móvel, fornecendo uma plataforma para manutenção e realização de testes de forma desligada dos servidores do serviço, potenciando a resolução de problemas que impedem as ligação à Internet, que anteriormente só poderiam ser resolvidos por técnicos especializados e após a desmontagem parcial da bicicleta.

1.4 SOLUÇÃO PROPOSTA

Ao longo deste documento, é identificada como limitação do sistema a existência de apenas uma interface de comunicação, o que limita a sua utilização às situações em que existe uma ligação de dados ativa. De forma a remover essa limitação, a solução proposta consiste na implementação de uma nova interface de comunicações, baseada em Bluetooth Low Energy.

Uma vez que, na ausência de uma ligação de dados, como numa manutenção da bicicleta que exija uma desmontagem parcial, não existe a possibilidade de realizar o diagnóstico através dos mecanismos existentes na plataforma de gestão centralizada das bicicletas que é parte integrante do sistema Bikeemotion, a solução proposta consiste no desenvolvimento de uma aplicação móvel que permita realizar um pedido de diagnóstico ao *hardware* da bicicleta.

1.5 ESTRUTURA

Esta dissertação encontra-se dividida em 8 capítulos.

O primeiro capítulo, dedicado à introdução ao tema, apresenta a motivação e objetivos por trás da mesma, assim como uma descrição da solução proposta e da organização do documento.

O segundo capítulo, dedicado ao estado da arte, aborda os conceitos fundamentais necessários para a contextualização do documento, assim como estabelece o panorama mundial do desenvolvimento de sistemas de gestão para serviços de partilha de bicicletas.

O terceiro capítulo, dedicado à apresentação do sistema Bikeemotion, estabelece um ponto inicial de desenvolvimento, descrevendo o sistema como se encontrava no momento em que se iniciou esta dissertação.

O quarto capítulo, dedicado à descrição do desenvolvimento, apresenta todo o trabalho realizado no âmbito desta dissertação.

O quinto capítulo é dedicado à aplicação móvel Android, apresentando em detalhe as várias partes que compõem a mesma, focando-se no desenvolvimento realizado neste trabalho.

O sexto capítulo é dedicado ao protocolo de comunicação desenvolvido para a troca de mensagens entre a aplicação móvel e o *hardware*.

O sétimo capítulo é dedicado aos testes e validação do desenvolvimento, apresentando os resultados obtidos e fazendo uma avaliação crítica do sucesso obtido com os desenvolvimentos.

O oitavo capítulo é dedicado às considerações finais, onde se fazem as conclusões e se refere o trabalho futuro.

Estado da Arte

O presente capítulo apresenta o nível atual da tecnologia nos pontos que são essenciais ao desenvolvimento deste trabalho. Em primeiro lugar, é feita uma apresentação da evolução dos sistemas de bicicletas partilhadas nos últimos anos, assim como os principais projetos existentes atualmente, sendo efetuada uma análise comparativa entre as características e funcionalidades que cada um desses sistemas apresenta. Para finalizar, é feita uma introdução às redes de área pessoal, no que diz respeito à sua arquitetura, características, requisitos e aplicações-alvo, sendo apresentadas várias tecnologias de comunicações existentes, dando-se mais ênfase às duas variantes da tecnologia Bluetooth, uma vez que correspondem às soluções mais indicadas para a implementação deste projeto. Neste âmbito, são apresentadas as versões 3.0, 4.0 e a versão de baixo consumo, o Bluetooth Low Energy, sendo estas apresentadas em detalhe, através da descrição das respetivas camadas.

2.1 SISTEMAS DE BICICLETAS PARTILHADAS

Um sistema de bicicletas partilhadas, também conhecido pela denominação inglesa *bike sharing system*, é um esquema de fornecimento de serviços, em que um conjunto de bicicletas é fornecido para uso público, durante um período de tempo curto, normalmente em viagens entre dois pontos, em que a bicicleta é recolhida num ponto inicial e devolvida num outro ponto, próximo ao final do percurso. Este tipo de serviços pode ser implementado com ou sem a intervenção de uma empresa privada, podendo ser fornecidos de forma gratuita ou requerendo um pagamento pelo aluguer. Muitos dos sistemas atualmente implementados fornecem assinaturas que fazem com que os primeiros 30 a 45 minutos de viagem sejam gratuitos ou muito baratos, encorajando o seu uso como meio de transporte e fazendo com que cada bicicleta sirva para vários utilizadores ao longo do dia. Em muitos destes sistemas, as aplicações móveis que acompanham os programas de bicicletas partilhadas implementados permitem mostrar, de forma instantânea e atualizada, as estações próximas com bicicletas disponíveis e docas livres.

Os sistemas de bicicletas partilhadas podem ser divididos em duas grandes categorias: programas de **Bicicletas Comunitárias**, organizados na sua maioria por grupos comunitários locais ou organizações sem fins lucrativos, e os programas de **Smart Bike**, implementados por agências governamentais, por vezes em parcerias publico-privadas. O objetivo principal desses sistemas é fornecer um acesso a bicicletas para viagens de curta distância nas áreas urbanas, como alternativa aos transportes públicos motorizados ou veículos privados, de forma gratuita ou de baixo custo, reduzindo assim o trânsito, ruído e poluição do ar. Estes sistemas também são reconhecidos como uma solução para o problema do *last mile* e a ligação dos utilizadores às redes públicas de transportes[4].

Muitos esquemas de partilha de bicicletas foram desenvolvidos por uma variedade de organizações ao longo dos anos, todos baseados num ou vários dos seguintes padrões de funcionamento:

- **Não regulado:**

Neste tipo de programa, as bicicletas são simplesmente disponibilizadas ao longo da cidade ou em áreas específicas da mesma para uso por qualquer pessoa. Nalguns dos casos, como *campus* universitários, as bicicletas são destinadas a ser usadas apenas dentro de uma zona geográfica restrita. Estas bicicletas são deixadas desbloqueadas numa área pública, como docas de bicicletas públicas, assim que o utilizador chega ao seu destino.

- **Depósito:**

Um pequeno depósito em dinheiro liberta a bicicleta de uma doca bloqueante e apenas volta a ser devolvido após a entrega da bicicleta num outro ponto autorizado. Este modo de funcionamento foi expandido para incluir o desbloqueio através de um cartão de crédito válido onde pudessem ser imputados os custos em caso de dano ou furto das bicicletas, depósitos de elevado valor e bloqueios de segurança obrigatórios.

- **Planos de assinatura:**

Neste tipo de sistema, existe uma identificação de membro, como um cartão de RFID¹, ou outro, que autentica o utilizador da bicicleta. Estes programas tentam reduzir o roubo das bicicletas exigindo aos utilizadores a compra de subscrições com cartões de crédito e equipando as bicicletas com sistemas de anti-roubo sofisticados e sensores de estado do veículo, cobrando o valor dos custos de reparação e substituição dos equipamentos no cartão de crédito do utilizador sempre que são detetados danos nas bicicletas ou quando estas não são entregues. Os serviços baseados em bicicletas elétricas tendem a seguir este programa, disponibilizando-as em estações de carga onde podem ser alugadas.

Este tipo de sistemas tende a ser implementado em parcerias publico-privadas ou com empresas que financiam o projeto em troca de publicidade nas estações de carga ou nas bicicletas, não sendo muito frequente encontrar projetos de partilha de bicicletas que sejam auto-sustentáveis no que se refere às suas receitas provenientes de assinaturas dos utilizadores.[5]

¹Identificação por Rádio-Frequência

2.1.1 Disponibilização das bicicletas ao utilizador

Os sistemas de bicicletas partilhadas podem disponibilizar os seus veículos através de duas grandes estratégias: baseadas em docas ou sem docas (com docas virtuais).

Os sistemas baseados em docas permitem alojar com muito mais segurança os veículos em estruturas especializadas, com uma maior robustez mecânica, que garantem a não remoção indevida por utilizadores não autorizados. A grande desvantagem de uma solução baseada em docas consiste no elevado investimento que é necessário realizar para implementar um grande número de docas, ao mesmo tempo que dificulta uma posterior reorganização da logística do sistema.

Por outro lado, os sistemas de bicicletas partilhadas que não são baseados em docas físicas correspondem a implementações em que as zonas de levantamento e entrega são delimitadas por coordenadas geográficas, não havendo uma infraestrutura física para o seu alojamento. Este tipo de solução permite uma reorganização rápida da localização dos veículos, bastando definir uma nova zona geográfica para alojar a doca virtual, resultando num sistema em que os custos de implementação são mais reduzidos.

Este tipo de sistemas tem como desvantagens a incapacidade de controlar as condições em que o utilizador devolve o veículo após a sua utilização, uma vez que o mecanismo de bloqueio é baseado numa algema com um curto alcance e o único parâmetro que é utilizado para a autorização da entrega corresponde à condição de se encontrar localizado dentro do perímetro geográfico definido pelas coordenadas GPS. As figuras 2.1 e 2.2 apresentam exemplos de situação em que a má conduta do utilizador provocou uma incorreta localização dos veículos.



Figura 2.1: Bicicletas deixadas em caixotes do lixo[6]

Na figura 2.1 é possível observar que a bicicleta foi deixada em cima de um caixote do lixo, um claro indício de que a localização GPS sozinha não permite identificar em que condições o veículo foi entregue.



Figura 2.2: Bicicletas depositadas no lago Merrit[7]

Por outro lado, na figura 2.2, observa-se um conjunto de bicicletas que foi depositada no lago Merrit, na Califórnia, Estados Unidos da América.

2.1.2 Soluções Comerciais

Em 1965, quando se dava, na Europa, o primeiro passo nas bicicletas partilhadas[8], estes sistemas eram muito rudimentares, com bicicletas que não se encontravam preparadas para um uso intensivo, o que levava a que se danificassem frequentemente. Este fator, aliado a serem dispostas livremente em certos espaços das cidades, para que pudessem ser facilmente utilizadas, levava a que fossem frequentemente furtadas, não levando a que existisse um enorme sucesso. À medida em que, a meados da década de 2000, com a introdução das tecnologias de informação, foram surgindo soluções mais robustas, colocadas em docas que resolviam os problemas do bloqueio e impedimento de roubos, foram sendo implementados novos sistemas, estando neste momento em clara expansão o número de sistemas implementados em todo o mundo, como se pode observar pela figura 2.3, divulgada pela NACTO², uma entidade que reúne os departamentos de transportes das cidades norte-americanas.

²National Association of City Transportation Officials

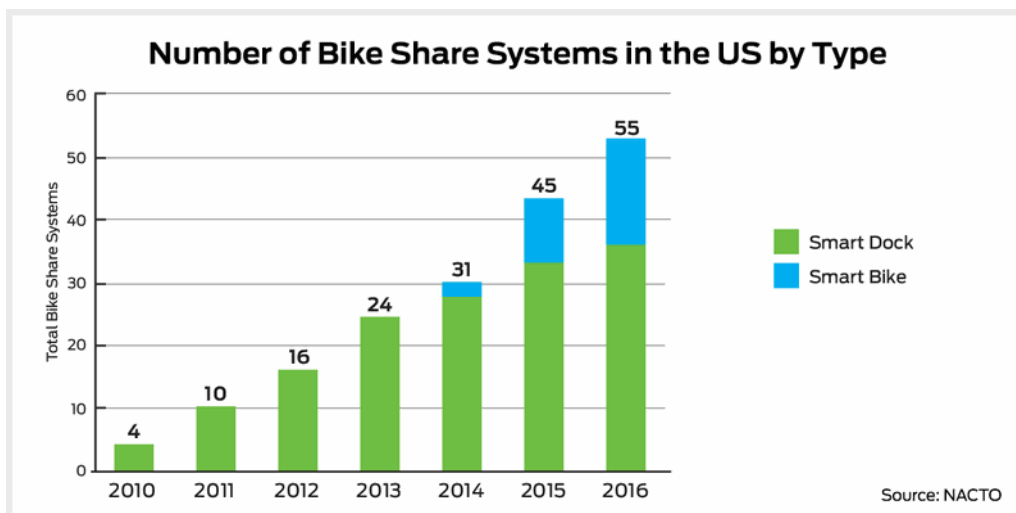


Figura 2.3: Evolução das bicicletas partilhadas por tipo nos EUA[9]

Segundo o estudo publicado pela NACTO[10], em 2016, ainda que sendo um valor marginal comparado com a totalidade da utilização dos transportes nos EUA³, no período entre 2010 e 2016 foram efetuadas cerca de 88 milhões de viagens em bicicletas partilhadas, partindo de um valor próximo de 0 em 2010 e atingindo cerca de 28 milhões de viagens em 2016, existindo cerca de 42000 bicicletas em circulação nos EUA nesse mesmo período, como demonstra a figura 2.4. Ao longo deste intervalo de tempo, evoluiu-se de 10 sistemas em 2010 para 55 em 2016, aparecendo espalhados ao longo do país, segundo a ilustração da figura 2.5.

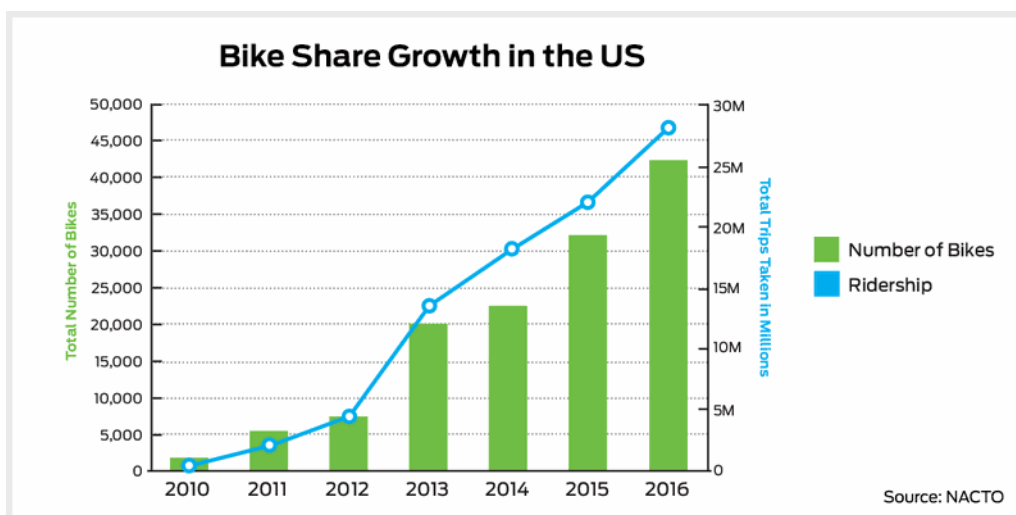


Figura 2.4: Número de bicicletas partilhadas e viagens realizadas entre 2010 e 2016[11]

³Estados Unidos da América

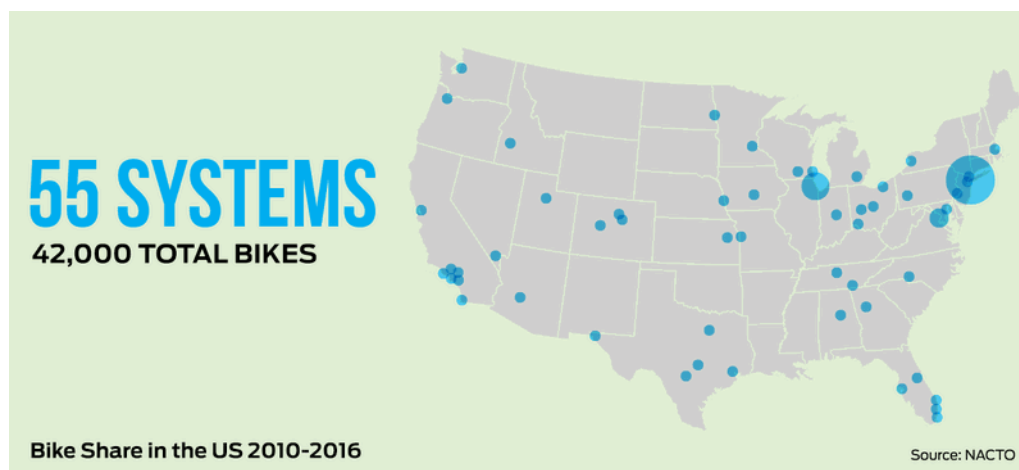


Figura 2.5: Mapa da distribuição dos sistemas de bicicletas partilhadas entre 2010 e 2016[12]

Derivado do aparecimento das novas soluções, foram sendo apresentadas novas funcionalidades entre concorrentes, o que permitiu obter uma diversificação das funcionalidades e características nos sistemas existentes em todo o mundo. Como algumas das soluções foram desenhadas com o objetivo da implementação em certas cidades em particular, nota-se nas características de alguns sistemas a orientação do seu desenvolvimento para um conjunto específico de características.

Ao contrário do que foi possível fazer com a solução da Bewegen, não foi possível obter informação de forma tão minuciosa acerca das implementações das outras soluções, pelo que a informação que é aqui apresentada não tem o mesmo nível de detalhe. Ao servir de base ao trabalho desta dissertação, o projeto da Bewegen foi profundamente analisado, obtendo-se todo o detalhe da implementação.

2.1.3 Bikeemotion

A Bikeemotion, empresa detentora do sistema de bicicletas partilhadas utilizado na solução Bewegen, surgiu em 2010 no seguimento de um projeto de IDT⁴ que reuniu o conhecimento e experiência de três empresas:

- **Micro I/O**

Empresa especializada em soluções tecnológicas e verticais, compostas pelo desenvolvimento integrado de componentes de hardware e software. Enquadrado no área das TICE⁵s, o conjunto de valências da empresa compreende o desenvolvimento de sistemas integrados de gestão de escolas, universidades e empresas, o desenvolvimento de soluções de eletrónica e *software* à medida, nas quais se inserem os controladores de máquinas de venda automática, controlo de acessos a instalações e cacifos, assim como o desenvolvimento de soluções no âmbito das cidades inteligentes, como gestão e monitorização de lugares estacionamento, gestão da recolha de lixo e monitorização atmosférica.

⁴Investigação e Desenvolvimento Tecnológico

⁵Tecnologias de Informação, Comunicação e Eletrónica

A Micro I/O é a empresa responsável pelo desenvolvimento e integração da componente de eletrónica do sistema e pelo desenvolvimento da componente de *firmware*;

- **Ubiwhere**

Empresa com grande experiência em aplicações móveis que têm em conta o contexto atual do utilizador, desde a sua posição geográfica e o reconhecimento de imagem até à gestão do perfil de utilizador.

A Ubiwhere é a empresa responsável pelo desenvolvimento de toda a componente de aplicações móveis;

- **Ponto.C**

Empresa especializada em soluções na área das Tecnologias de Informação, integrando as divisões de Software, Consultoria e Redes com uma grande capacidade para o desenvolvimento de novos produtos e soluções.

A Ponto.C é a empresa responsável pelo desenvolvimento do sistema de gestão, a componente integradora que permite gerir todo o sistema de partilha de bicicletas.

O sistema Bikeemotion é constituído pelo equipamento eletrónico que pode ser embutido em bicicletas tradicionais ou eletricamente assistidas, página web, aplicação móvel para gestão do utilizador, quiosque de entrega e carregamento de bicicletas, assim como o registo e interface com o utilizador, e plataforma de gestão, para administração do sistema. As bicicletas são equipadas por diversos sensores de atividade que permitem saber o estado de cada bicicleta em tempo real, enquanto que o sistema integrado de gestão permite o controlo e administração de todo o sistema.

Atualmente, a Bikeemotion é o parceiro tecnológico exclusivo da Bewegen Technologies Inc.[13], tendo já implementado os sistemas de partilha de bicicletas em várias localizações[14], como se observa na tabela 2.1.

Continente	Localização	Bicicletas	Estações	Estado
América do Norte	Birmingham	400	40	implementado
	Baltimore	500	50	implementado
	Howard County	78	9	implementado
	Summit County	88	9	implementado
	Richmond	220	20	implementado
Europa	Wolfsburg	50	10	implementado
	Lagoa	30	3	implementado
	Barreiro	10	2	implementado
	Rio Maior	24	5	implementado
	Coruche	27	4	em implementação

Tabela 2.1: Sistemas atualmente implementados pela Bikeemotion

Uma vez que o sistema apresentado neste capítulo serve de base ao desenvolvimento desta dissertação, o detalhe apresentado na descrição desta solução será claramente superior ao utilizado para as soluções apresentadas no capítulo 3. Da mesma forma, uma vez que será detalhada a arquitetura do sistema Bikeemotion nesse capítulo, esta secção funciona como uma descrição abrangente e sintética do sistema.

As bicicletas elétricas utilizadas pela Bewegen foram desenvolvidas em parceria com o fabricante Procycle, ilustradas na figura 2.6. Derivado de todo o equipamento que é necessário alojar na estrutura metálica das bicicletas, estas tiveram que ser adaptadas, por forma a ser possível essa integração. Esta integração foi possível, uma vez que se reuniram os esforços de todas as equipas presentes no desenvolvimento da eletrónica e mecânica das bicicletas, promovendo a simbiose entre todos os componentes. Estas bicicletas são equipadas com baterias de lítio de 600 Wh, numa configuração de 48 V, contendo sensores de torque e/ou de rotação da pedaleira, permitindo uma análise do esforço exercido pelo utilizador, um motor elétrico que atua em auxílio do utilizador, funcionando como um reforço do ritmo do pedalar do utilizador, diminuindo a força que este tem que impor à bicicleta, e um conjunto de duas placas eletrónicas de controlo, montadas em forma de sanduíche, funcionando como controladoras de todo o funcionamento da bicicleta, interface com o utilizador, comunicação com os servidores, bloqueio e desbloqueio das bicicletas e localização dos equipamentos em tempo real, assim como a obtenção dos dados do estado de cada ponto do sistema.



Figura 2.6: Bicicleta Bewegen[15]

Por forma a dotar as bicicletas de uma capacidade de controlo sobre a atuação dos motores, com base nos dados do estado do equipamento e da atuação do utilizador exercida sobre ele, obtidos através dos sensores, foi necessário desenvolver um conjunto de eletrónica adicional, que fosse possível embutir na estrutura metálica da bicicleta, permitindo a unificação do sistema após serem efetuadas todas as alterações necessárias para permitir essa integração. A eletrónica implementada consiste numa placa-mãe, tendo como principal objetivo concentrar o

poder de processamento, realizar decisões em relação aos momentos e intensidade da assistência ao utilizador, assim como gerir a localização GPS⁶ e efetuar a interface entre os sensores da bicicleta e a unidade de interface com o utilizador, que se encontra integrada numa segunda placa, que concentra a comunicação 3G⁷ através de um modem, a leitura de cartões de RFID e um ecrã LCD⁸. A interligação entre as duas placas é conseguida através de dois conectores de 24 pinos existentes em ambas as placas.

O controlo da assistência ao utilizador é realizado pela placa-mãe, com base nos dados obtidos pelos sensores de torque ou velocidade, ao mesmo tempo que são tidas em conta as definições reportadas pelo servidor através da ligação de dados 3G ou GPRS⁹, caso o sistema esteja configurado para ter uma ligação de dados. Com base em todos esses dados, é possível perceber em que condições se encontra o utilizador, fornecendo a esta placa controladora toda a informação necessária para tomar as decisões sobre os momentos chave para a atuação dos motores.

O sistema Bikeemotion permite o funcionamento em modo de doca física ou em doca virtual, sendo que o mecanismo de bloqueio é baseado em cadeado bloqueado eletronicamente, permitindo ser atracado de forma mais segura a vários elementos da infraestrutura municipal.

2.1.4 BIXI Montréal

Lançado em Maio de 2009 em Montreal, Canadá, foi o primeiro sistema de bicicletas partilhadas norte-americano e o sistema original da marca BIXI, implementado pela PBSC, uma empresa criada pela cidade de Montreal para fornecer o sistema. Após a falência da PBSC, no início do ano 2014, a cidade de Montreal comprou os ativos existentes na cidade e criou a BIXI Montréal, uma entidade sem fins lucrativos para as operações do sistema de bicicletas partilhadas, enquanto que os restantes ativos das implementações internacionais foram transferidas para uma nova empresa, agora com o nome PBSC Urban Solutions.

Surgindo o nome da união das palavras *bicycle* e *taxi*, para relevar o conceito de ser possível utilizar uma bicicleta da mesma forma que um táxi, este sistema é constituído pelas estações de pagamento, as bicicletas e as docas de bloqueio, onde as bicicletas são mantidas, tendo este sistema a característica de ser modular e alimentado por painéis solares.

As estações de ancoragem são formadas pela combinação de grupos de quatro docas, aproveitando a sua estrutura modular, o que permite que uma estação de pagamento seja colocada no espaço de uma doca. Esta estrutura modular permite também uma melhoria na capacidade de reparação e manutenção, na medida em que um módulo pode facilmente ser substituído por um idêntico, reduzindo os tempos de em que o sistema se encontra em baixo. As estações de pagamento permitem ainda alugar uma bicicleta utilizando uma chave de subscrição, obtida através de uma subscrição de longo termo online ou de um código de acesso obtido numa estação de pagamento. Estes pontos são operados através de um ecrã tátil e apenas suportam cartões de crédito, tendo como característica adicional a presença de

⁶Global Position System

⁷3ª Geração de Redes Móveis

⁸Liquid Crystal Display

⁹General Packet Radio Service

um botão que permite notificar a equipa técnica da necessidade de manutenção de uma ou mais bicicletas.

As bicicletas foram desenhadas com o intuito de serem uni-sexo, com um quadro *step-through* e uma caixa de 3 velocidades. O quadro em alumínio de peça única permite esconder os cabos com o objetivo de os proteger das intempéries e da manipulação por terceiros.



Figura 2.7: Docas e quiosque Bixi[16]

A figura 2.7 apresenta a disposição habitual nas docas de armazenamento e o quiosque de aluguer e pagamento das viagens. É possível observar, também, que esta disposição permite minimizar o espaço ocupado por todo o sistema.

2.1.5 BiciMad

O projeto BiciMad foi implementado na cidade de Madrid, Espanha, em 2014, e consiste num novo serviço de transporte público prestado em 100% por bicicletas elétricas, de forma fácil, ecológica e sem dificuldades. Este projeto de bicicletas partilhadas coloca à disposição do utilizador um conjunto de 2028 bicicletas, dispostas pela cidade, 165 quiosques de pagamento e 4116 âncoras de estacionamento, levantamento e entrega de bicicletas.

Por forma a melhorar a experiência do utilizador, este projeto emprega as tecnologias de informação e comunicação, permitindo a consulta da informação das bicicletas e das estações disponíveis, através de um portal na Internet ou de uma aplicação móvel, ao mesmo tempo que permite a reserva instantânea mediante a associação de um cartão de crédito ou de débito.

O levantamento e a devolução das bicicletas é um processo rápido e sem complicações, sendo que se encontram disponíveis dois planos de utilização: o plano anual, cujo registo pode ser feito presencialmente num posto de informações, através da plataforma web ou da aplicação móvel disponibilizada pelo serviço, que concede ao utilizador o acesso a um cartão de desbloqueio das bicicletas enquanto este não esgotar o número de viagens que ainda tem disponíveis, e o plano ocasional, onde é possível adquirir um cartão de desbloqueio das bicicletas válido por 1, 3 ou 5 dias.

Para se utilizar este serviço, o utilizador apenas tem que inserir o cartão numa das estações

de levantamento de bicicletas e removendo uma das bicicletas que se encontrar com o LED¹⁰ de cor verde, podendo remover a bicicleta após este passo.



Figura 2.8: Interface com o utilizador das bicicletas BiciMad[17]

As bicicletas escolhidas para este projeto correspondem a peças da marca Booster Bike. Estes equipamentos correspondem a versões de bicicletas elétricas simplistas, cujo modo de funcionamento auxilia o seu utilizador ao pedalar, inserindo-se na categoria das bicicletas Pedelec. Estas bicicletas contêm na sua direção um conjunto de interfaces com o utilizador, ilustrados pela figura 2.8, das quais se destacam um botão vermelho, que permite ligar e desligar a bicicleta, um botão verde, que permite ligar e desligar os faróis dianteiros, um botão verde, que permite regular o nível de assistência ao utilizador, entre três níveis (alto, médio ou baixo), e um indicador no nível de bateria, apresentado na forma de uma rampa no lado direito do comando de controlo.

A aplicação móvel que acompanha o uso da bicicleta permite estabelecer rotas, com o auxílio da navegação por GPS, obter a meteorologia para a cidade de Madrid, obter informações sobre o histórico da conta do utilizador, como o saldo, viagens anteriores, estimativa das calorias consumidas e da quantidade de CO_2 poupada com o uso deste sistema. Aliada a toda esta informação, é ainda possível consultar o número de lugares que se encontram disponíveis nas estações de entrega de bicicletas.

2.1.6 Bycyklen

O projeto de bicicletas partilhadas Bycyklen foi implementado na cidade de Copenhaga, Dinamarca, no ano 2014 e consiste num conjunto de 1860 bicicletas elétricas e 90 estações de levantamento e entrega, disponíveis a qualquer utilizador, não se limitando aos planos de longa duração. Através destas estações espalhadas pela cidade, é possível fazer a recolha de uma bicicleta, introduzindo na interface tátil da bicicleta o nome de utilizador e respetivo código, e iniciando a viagem.

O painel tátil que serve de interface ao sistema disponibiliza um mecanismo de gestão de rotas e navegação por GPS, como é ilustrado na figura 2.9, onde se pode estabelecer um

¹⁰Light Emitting Diode

caminho para um ponto de interesse, obtido através da rede móvel existente no equipamento, e que pode ser guardado para futura reutilização. Associada a esta funcionalidade, é possível também verificar o estado da bateria e ajustar o nível de assistência do motor elétrico, sendo que este irá apenas atuar até que seja atingida a velocidade máxima de segurança de 22 Km/h, tendo uma autonomia máxima de 25 Km, dependendo do tipo de condução do utilizador.



Figura 2.9: Interface tátil com o utilizador das bicicletas Bycyklen[18]

A devolução das bicicletas pode ser assistida pela navegação GPS que fornece a localização de todas as estações que se encontram próximas, assim como fornece informação acerca do número de docas vazias que existem para receber devoluções de bicicletas.

As bicicletas escolhidas para equipar este projeto foram fornecidas pela GoCycle, sendo pertencentes ao modelo GoBike. Este modelo contém um quadro resistente, capaz de transportar um utilizador e respetiva carga até 100 Kg, estando também equipada com um motor elétrico de 250W e uma bateria de 300 Wh de 22V. A definição de assistência base do motor encontra-se no nível 2, numa escala que inclui os valores de 0 a 5, em que 0 corresponde ao motor não efetuar assistência e 5 a corresponder à assistência máxima do mesmo. Em qualquer um destes níveis, o valor da velocidade máxima em que é possível assistir mantém-se inalterado.

Todas as bicicletas têm uma tranca eletrónica na roda traseira, que permite ao utilizador estacionar a bicicleta durante a sua viagem. Para proceder ao bloqueio, a interface gráfica fornece um conjunto de passos após os quais apenas o utilizador poderá voltar a desbloquear a bicicleta, mantendo-se a cobrança do uso do sistema durante o tempo em que ela estiver bloqueada fora da doca de devolução.

Os planos de contratação de serviços existentes neste sistema são bastante flexíveis quanto à duração da utilização das bicicletas, sendo divididos nos seguintes casos:

- O plano de utilização à medida do necessário é a solução apresentada para utilizadores esporádicos, com um custo de 30 DKK¹¹ por cada hora, o equivalente a 4€.

¹¹Coroa Dinamarquesa

- O plano mensal inclui 140 minutos de viagem, com um custo de 70 DKK, o equivalente a 9,41€, sendo que cada minuto adicional fora do plano é cobrado a 0.35 DKK, o equivalente a 0,05€, e cada minuto em modo de bloqueio é cobrado a 0.15 DKK, o equivalente a 0,02€.
- Os pacotes anuais de 10 horas por 300 DKK, equivalente a 40€, e de 20 horas por 500 DKK, equivalente a 67,22€, permitem a utilização do plano por 5 utilizadores simultâneos e têm a validade de um ano. Cada minuto adicional fora do plano é cobrado a 1 DKK, equivalente a 0,13€.

É possível reservar uma bicicleta com antecedência, sendo que esse procedimento está sujeito a uma taxa aplicável que corresponde a 5 DKK, equivalente a 0,67€. Meia hora antes do momento previsto para o levantamento do veículo, o sistema envia ao utilizador um lembrete com um código que permite o cancelamento da reserva de forma gratuita. Se o utilizador optar por não cancelar, a taxa de reserva passa para o valor de 15 DKK, o equivalente a 2,01€.

2.1.7 ESB

O sistema ESB apresenta-se como sendo um mecanismo genérico para ser implementado num conjunto de modelos de bicicletas de marcas como Bulls, e-flow, diavelo e A2B, não sendo uma implementação largamente existente em projetos de bicicletas públicas à escala de cidades, fornecendo um serviço que pode ser implementado por terceiros.

Este sistema consiste numa aplicação móvel que se baseia numa ligação sem fios BLE para comunicar com a placa controladora, um dispositivo vendido discretamente, para ser implementado em soluções do cliente. Esta placa pode ser adquirida também com um módulo de comunicação GPRS, caso seja necessário comunicar com um servidor. A aplicação móvel é o mecanismo primordial de comunicação com o servidor, fornecendo geolocalização da bicicleta, chamadas de emergência *e-call* quando existe um acidente, diagnóstico remoto, navegação por GPS e partilha das rotas nas redes sociais, como é ilustrado na figura 2.10.

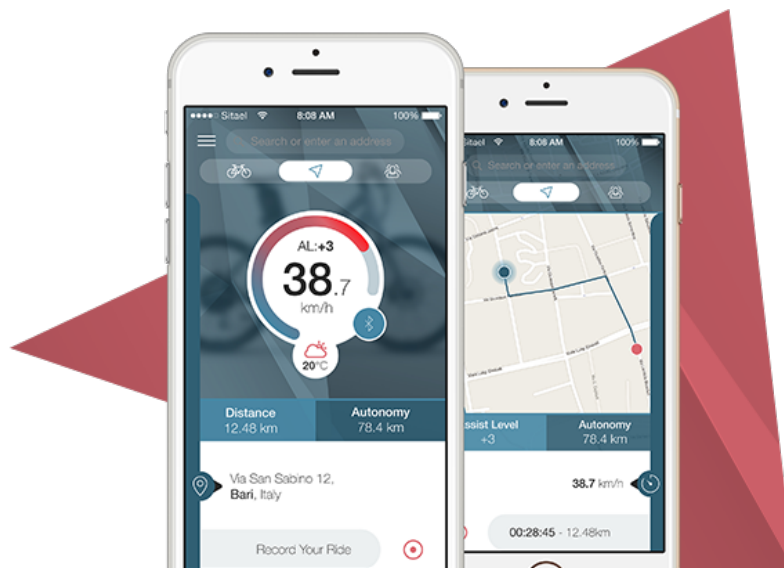


Figura 2.10: Aplicação Móvel ESB[19]

O sistema de controlo de bateria pode ser adquirido separadamente, uma vez que apenas é implementado nas soluções que contenham bicicletas elétricas, sendo que os planos de serviço existentes não contemplam esse controlo, estando apenas disponível caso seja contratado o serviço de gestão de frota B2B¹².

Os planos de serviço disponíveis contemplam a deteção de acidentes, roubos e verificação remota do estado das bicicletas, no caso da versão *plus*, estando o resto das funcionalidades apresentadas acima disponível em ambos os planos. Associado a estes planos, cujo objetivo é permitir um sistema para um pequeno número de bicicletas, este sistema dispõe ainda de uma vertente B2B, em que o controlo é implementado na nuvem, sendo possível obter todos os dados existentes na aplicação, mas num ambiente centralizado, numa lógica de armazenamento na nuvem.

Com este perfil, é fornecida uma plataforma para gestão da frota que permite traçar o perfil de uso de cada bicicleta e que contém ferramentas de gestão que auxiliam a logística de distribuição das bicicletas pelos pontos de levantamento existentes. Além destas funcionalidades, é possível obter a geolocalização da frota e obter informação sobre o comportamento dos utilizadores, avaliando as condições dos veículos em tempo real, ao mesmo tempo que permite exportar os dados para serem usados por outras aplicações.

2.1.8 Velobility

O sistema Velobility, lançado pela empresa Suíça SwissFleet, consiste numa integração de todos os componentes necessários para a implementação de um serviço de bicicletas partilhadas, restringindo todos esses componentes a uma caixa que é embutida na zona da caixa de velocidades da bicicleta, como é ilustrado na figura 2.11. Este sistema efetua toda a gestão do estado da bicicleta, podendo ser inserido em bicicletas tanto do tipo elétrico como

¹²Business to Business

nas tradicionais. Este sistema contém ainda todos os componentes necessários para estabelecer uma ligação através da rede móvel aos servidores do fornecedor do serviço, permitindo a sua implementação em soluções baseadas em computação na nuvem.



Figura 2.11: Encapsulamento dos componentes mecânicos e eletrónicos[20]

Por forma a melhorar a experiência do utilizador, as interfaces para com este foram desenvolvidas de forma a permitir a personalização de alguns aspetos do sistema. Uma das personalizações principais consiste no mecanismo de autenticação do utilizador, que permite que seja feito através de identificação por NFC¹³, cartões de RFID, desbloqueio através de uma aplicação móvel ou da introdução de um código no teclado à prova de água que pode ser embutido neste sistema.

Uma das questões que levantou mais preocupações aos criadores deste sistema foi a questão do vandalismo. Para combater este problema, foi desenvolvido um mecanismo de bloqueio do pé de descanso da bicicleta, impedindo que esta possa ser conduzida por outra pessoa que não o utilizador, nas pausas intermédias do percurso, durante o período de tempo em que o veículo está atribuído a um utilizador. Aliado a isto, o mecanismo de encapsulamento único impede o acesso aos componentes e, por conseguinte, o seu desbloqueio ilícito.

Atualmente, este sistema encontra-se implementado em cidades como Biel[21], Zurich[22], Geneva[23], Lausanne[24], Kassel[25], Mainz[26], Bern e Thun[27] e Winterthur[28].

2.1.9 Lidl Bike

Lançado em março de 2017, o sistema Lidl Bike é uma solução de bicicletas partilhadas lançada pela DB Rent, na cidade de Berlim, Alemanha, em conjunto com o patrocinador, a cadeia de hipermercados Lidl. Este sistema conta com 3500 bicicletas disponíveis para o utilizador em toda a cidade, fazendo parte da família Call a Bike, atualmente o maior sistema de bicicletas partilhadas da Alemanha.

Existem pontos de devolução das bicicletas em todas as principais estradas da cidade, como os exemplificados na figura 2.12, garantindo-se um destes pontos no máximo a 200 metros de cada uma das 39 lojas Lidl existentes nas imediações.

¹³Near Field Communication



Figura 2.12: Zona de parque do sistema Lidl Bike[29]

Uma vez que se trata da mesma família de sistemas, um utilizador do sistema Call a Bike pode utilizar o sistema Lidl Bike sem ter um registo adicional, acontecendo o mesmo no sentido contrário, em que um utilizador do sistema Lidl Bike pode utilizar uma bicicleta do sistema Call a Bike com a mesma identificação. As tarifas utilizadas correspondem a 1 euro por cada 30 minutos, sendo que o primeiro período utilizado é gratuito.

2.1.10 Análise Comparativa das soluções

A tabela 2.2 apresenta uma comparação entre as funcionalidades existentes em cada sistema, oferecendo um resumo das capacidades dos mesmos.

No que se refere à interface com o utilizador, o sistema fornecido pela Bycyklen é o que apresenta o melhor nível de funcionalidades, uma vez que oferece ao utilizador, de forma integrada e transparente, num ecrã tátil, todos os serviços disponíveis na bicicleta, incluindo a navegação por GPS. Ainda com um bom nível de interação com o utilizador, ainda que com menos funcionalidades facilmente acessíveis na bicicleta, o sistema fornecido pela Bikeemotion permite o acesso a valores de velocidade instantânea e distância percorrida através de um pequeno ecrã a cores existente no centro do guiador. Os sistemas da Velobility e da ESB apenas permitem a interação do utilizador com a bicicleta através da aplicação móvel, ainda que seja possível desbloquear a bicicleta fisicamente no *hardware*, no caso da Velobility. No caso da Bicimad, a interação com o utilizador é muito reduzida, ainda que a experiência seja complementada pela aplicação móvel. Derivado da ausência do motor elétrico, as bicicletas Bixi não têm qualquer interface com o utilizador, que não sejam as mecânicas para controlo da campainha e engrenagem de redução de velocidades.

Exceto as bicicletas da Bixi que apenas têm esse mecanismo quando na doca, todos os sistemas abordados, apresentam mecanismos de segurança que impedem o roubo dos veículos, mas apenas os sistemas da Bikeemotion, Bycyklen e ESB, na sua versão de B2B com gestão de

frota, permitem a localização em tempo real das bicicletas, o que pode ser um fator decisivo no momento de tentar recuperar os veículos após o furto. Aliado a isto, todas incluem bloqueios mecânicos através de trancas, sendo de relevar o bloqueio do pé de descanso, existente no sistema da Velobility.

Os sistemas da Bikeemotion (BE¹⁴), Bicimad e Bycyklen são os que apresentam um maior número de bicicletas elétricas em utilização no terreno, enquanto que o sistema da Velobility é o que se apresenta no maior número de cidades. No caso do sistema da ESB, não é fornecida informação acerca do número de implementações atualmente existentes.

Função	BE	Bixi	ESB	BiciMad	Bycyklen	Velobility
Localização por GPS	X	—	X	—	X	X
Navegação por GPS	—	—	—	App	X	—
Assistência de motor configurável	X	—	—	X	X	X
Bloqueio anti-furto	X	X	X	X	X	X
Sistema de docas modular	X	X	—	X	X	—
Desbloqueio por RFID	X	X	—	X	—	X
Desbloqueio por PIN	—	X	—	—	X	X
Aplicação Móvel	X	X	X	X	—	X
Interface tátil	—	—	—	—	X	—
Deteção de acidentes	X	—	—	—	X	—
Deteção de mau uso da bicicleta	X	—	—	—	—	—
Ligação 2G/3G	X	X	Opcional	—	X	X
Disponível todo o ano	X	—	Opcional	X	X	X
Compatível com bicicletas elétricas	X	—	X	X	X	X
Compatível com bicicletas tradicionais	X	X	X	—	—	X

Tabela 2.2: Tabela das funcionalidades disponibilizadas por cada sistema

Excluindo o caso do sistema da Bycyklen, cuja interface inclui todas as funcionalidades que o fabricante pretendia ver implementadas no sistema, todas as outras soluções incluem uma aplicação móvel que serve de complemento às funcionalidades nativas das bicicletas, adicionando novas características que compensam algumas das lacunas em termos de funcionalidades das mesmas, tal como a navegação por GPS ou a indicação da velocidade e distância percorridas. No caso dos sistemas da Velobility e ESB, a aplicação móvel consiste

¹⁴Bikeemotion

em praticamente toda a interface existente entre o utilizador e a bicicleta. Assim sendo, todas as funcionalidades existentes estão implementadas na aplicação móvel. No caso da Bicimad, a pobre interface com o utilizador existente na bicicleta é compensada pelas funcionalidades de navegação por GPS e localização de pontos de interesse existentes na aplicação móvel.

2.2 REDES DE ÁREA PESSOAL

Uma rede de área pessoal, conhecida pela denominação inglesa PAN¹⁵, define-se como uma rede de comunicações utilizada para transmissão de dados entre equipamentos, com um alcance relativamente curto, normalmente até 10 metros, em que um deles assume o papel de mestre enquanto que os outros adquirem o papel de escravos. Segundo esta definição, considera-se uma rede de área pessoal sem fios, WPAN¹⁶, como sendo uma PAN de baixo consumo sobre uma tecnologia de redes sem fios de curto alcance, como uma ligação por infravermelhos, Bluetooth Classic ou Bluetooth Low Energy, destinada a substituir os cabos tradicionalmente existentes nas ligações entre o computador e os periféricos.

As redes celulares, assim como as redes de área local sem fios e as redes de área pessoal, têm tido um grande impacto na sociedade, ao apresentarem uma maior mobilidade e flexibilidade, oferecendo um menor custo e uma mais rápida instalação e adaptação da rede, quando comparadas com uma solução que envolva a passagem de cabos. As redes sem fios podem ser classificadas segundo a sua área de abrangência em: redes pessoais, WPAN, para distâncias muito curtas, redes locais, WLAN¹⁷, para distâncias na ordem das poucas centenas de metros, redes metropolitanas, WMAN¹⁸, para distâncias que abrangem uma cidade, e as redes geograficamente distribuídas, WWAN¹⁹, para distâncias que abrangem várias cidades, países ou continentes.

Entre as tecnologias de redes de comunicações sem fios, as WPAN são as que mais têm tido destaque nos últimos anos devido à grande revolução tecnológica de que se tem assistido. As redes sem fios de área pessoal (WPAN) são utilizadas para troca de informação entre dispositivos portáteis a distâncias relativamente pequenas e que requerem normalmente um gasto de energia reduzido. Ao contrário das redes de área local (WLAN), as WPAN não requerem infraestruturas de rede muito complexas, na medida em que o alcance das comunicações é claramente inferior ao das WLAN, pois o objetivo é cobrir a área em torno do utilizador e não servir múltiplos utilizadores. Cada vez existem mais dispositivos móveis de carácter pessoal que assumem um papel importante no dia-a-dia do utilizador, tornando a interligação entre dispositivos mais cómoda e menos dispendiosa, face aos dispositivos com fios, tornando assim possível desenvolver soluções mais económicas para um maior espectro de aparelhos.

¹⁵Personal Area Network

¹⁶Wireless Personal Area Network

¹⁷Wireless Local Area Network

¹⁸Wireless Metropolitan Area Network

¹⁹Wireless Wide Area Network

Desde a última década, existe um grupo de trabalho do IEEE²⁰, denominado 802.15, que se dedica a trabalhar na especificação e desenvolvimento de tecnologias de redes de área pessoal sem fios, sendo que atualmente existem diversos subgrupos de trabalho, cada um deles focado em diferentes vertentes dentro do conceito geral das WPAN. O desenvolvimento contínuo de normas gerais para as tecnologias é importante, pois garante a interoperabilidade de diversos dispositivos, reduzindo assim a complexidade e os custos para o utilizador final.

Como foi referido acima, o IEEE baseou-se na primeira versão do Bluetooth para desenvolver o standard IEEE 802.15.1. Esta tecnologia, especificada pelo Bluetooth SIG²¹, mostrou enorme potencial sendo atualmente um dos exemplos mais comuns das WPAN. O Bluetooth SIG é formado por nomes sonantes do sector privado ligados às comunicações, como Ericsson, Nokia ou Motorola, e que, após rutura com o IEEE, continuou o desenvolvimento da tecnologia até chegar à versão 5.0, a versão atual, que inclui um novo sistema de comunicações com atenção acrescida para os gastos energéticos, designado por BLE. Este novo subconjunto da última versão do Bluetooth é útil para aplicações que requerem baixo débito de informação mas que possam atuar durante longos períodos de tempo, sem a necessidade de substituição de bateria, e que se apresenta com uma grande vantagem, a de estar incorporado na maioria dos *smartphones* atuais, o que não acontece com outras tecnologias, como o ZigBee [30], que também é conhecido por ser uma tecnologia de baixo consumo energético. O aparecimento desta vertente tecnológica veio fomentar ainda mais o uso de sensores nas chamadas redes de sensores sem fios, WSN²². Nestas redes de sensores, estas tecnologias de comunicação são normalmente usadas em aplicações para monitorizar estados ou comportamentos e enviar essa pequena quantidade de informação para uma estação central, onde esta é tratada. Neste caso, a estação central pode ser constituída por um *smartphone* atual com sistema operativo Android e suporte para BLE, uma vez que este é um sistema operativo móvel com grande abrangência no segmento dos dispositivos móveis e que disponibiliza o suporte para o desenvolvimento de aplicações que utilizam estes recursos de comunicação.

Derivado da sua massiva implementação na quase totalidade dos dispositivos móveis existentes atualmente, o Bluetooth, e a sua versão de baixa energia, são as soluções que mais se adequam ao desenvolvimento deste trabalho, na medida em que permitem que não seja necessário equipamento adicional para a utilização de outras interfaces de comunicação, assim como reduz o esforço de desenvolvimento dessas interfaces, derivado da implementação nativa destas tecnologias no SDK²³ dos sistemas operativos.

2.2.1 ZigBee

ZigBee[30] é uma especificação baseada na norma IEEE 802.15.4, definida em 1998 e padronizada em 2003, destinada para protocolos de comunicação de alto-nível usados para criar PAN's com pequenos dispositivos rádio de baixo consumo, em soluções como domótica, dispositivos médicos de captura de dados ou outros projetos que necessitam de ligações sem

²⁰Institute of Electrical and Electronics Engineers

²¹Special Interest Group

²²Wireless Sensor Network

²³Software Development Kit

fios com baixa largura de banda e baixo consumo. A tecnologia definida pela especificação ZigBee tem como objetivo ser mais simples e menos dispendiosa que outras WPAN, como o protocolo Bluetooth ou o Wi-Fi.

O seu baixo consumo energético limita as suas distâncias de transmissão a cerca de 10 a 100 metros em linha de vista, dependendo da potência de saída. Apesar disto, os dispositivos ZigBee conseguem atingir maiores distâncias fazendo uso de redes de comunicações em malha que vão transmitindo a informação entre os nós até atingir o destino. Esta tecnologia é tipicamente utilizada em aplicações de baixa taxa de dados que requerem uma autonomia elevada e comunicações seguras, visto que as redes ZigBee são seguras com chaves simétricas de 128 bit.

O ZigBee utiliza as bandas rádio ISM²⁴, nomeadamente a banda dos 2.4 GHz e dos 868 MHz, na Europa, e suporta de forma nativa as redes em estrela e em árvore, assim como as redes em malha genéricas. Cada rede deve ter um dispositivo coordenador, cujas tarefas são a de criar, gerir e controlar os parâmetros da rede. Nas redes em estrela, o coordenador deve ser o nó central, enquanto que nas redes em árvore e em malha, podem ser utilizados *routers* ZigBee para estender a comunicação ao nível da rede.

As redes ZigBee são compostas pelos seguintes três tipos de dispositivos, que podem diferir apenas na função que lhes foi atribuída ou também no conjunto de funcionalidades e fonte de alimentação permitidas:

- **Coordenador:**

Este é o dispositivo com o máximo de funcionalidades, forma a raiz da árvore da rede, sendo capaz de servir como interface para outras redes. Existe apenas um coordenador ZigBee em cada rede, apesar de ser possível estabelecer uma rede sem um coordenador, sendo esta função atribuída ao dispositivo que iniciou a rede (o primeiro a ser ligado). Este elemento funciona também como Ponto de Confiança e repositório para as chaves de segurança.

- **Router:**

Ao mesmo tempo que desenvolve uma função aplicacional, um *router* pode atuar como *router* intermédio, encaminhando dados de outros dispositivos.

- **Dispositivo terminal:**

Contém apenas as funcionalidades necessárias para comunicar com um pai (um coordenador ou um *router*), não conseguindo enviar dados para outros nós. Esta relação permite que o nó esteja a dormir durante um período significativo de tempo, o que garante uma maior autonomia.

2.2.2 Ligação por Infravermelhos

Uma ligação por infravermelhos consiste na utilização de um díodo emissor de radiação infravermelha para a comunicação entre dois dispositivos em linha de vista. Esta ligação é

²⁴Industrial, Scientific and Medical

essencialmente bidirecional mas apenas um dispositivo pode comunicar de cada vez, uma característica denominada pela designação inglesa *half-duplex*.

A norma *IrDA*²⁵ é a utilizada atualmente para ligações por infravermelhos, que tem como características a transferência de dados fisicamente segura, em linha de vista e com uma taxa de erros reduzida. Esta norma permite ligações dados desde 2.4 kbit/s até 1 Gbit/s e um alcance de 1 m.[31]

2.2.3 Bluetooth Classic

Bluetooth é uma tecnologia sem fios para a troca de dados em curtas distâncias entre terminais fixos e móveis, construindo redes PAN. Desenvolvida em 1994 pela Ericsson, foi inicialmente concebida para a se tornar uma alternativa aos cabos RS-232, sendo atualmente gerida pelo Bluetooth SIG.

Bluetooth opera nas frequências entre 2402 e 2480 MHz, ou 2400 e 2483.5 MHz se forem incluídas as bandas de guarda inferior de 2 MHz e superior de 3.5 MHz, que pertencem ao grupo das gamas de frequências não licenciadas e de uso livre, mais precisamente à gama de frequências ISM. A tecnologia de rádio usada pela norma Bluetooth é a de espalhamento de espectro por salto na frequência, conhecido pela denominação inglesa FHSS²⁶, que consiste no envio dos dados através da prévia divisão em pacotes que são enviados num dos 79 canais definidos pela norma Bluetooth, sendo o canal escolhido para o envio alterado 800 vezes por segundo.

Originalmente, o esquema de modulação GFSK²⁷ era o único disponível, mas com a introdução da norma 2.0 com EDR²⁸, passaram a poder ser utilizados os esquemas de modulação $\frac{\pi}{4}$ -DQPSK²⁹ e 8-DPSK³⁰ entre dispositivos compatíveis. O primeiro esquema de modulação referido deriva do esquema DQPSK tradicional, em que toda a constelação sofre uma rotação de $\frac{\pi}{4}$ em relação à sua posição original[32].

Os dispositivos que comunicam utilizando a modulação GFSK original são descritos como trabalhando no modo BR³¹, apenas conseguindo atingir uma taxa de transferência teórica de 1 Mbit/s, enquanto que o uso dos esquemas de modulação $\frac{\pi}{4}$ -DQPSK e 8-DPSK permitem uma taxa de transferência teórica de 2 Mbit/s e 3 Mbit/s, respetivamente, sendo designados como funcionando no modo EDR. Com a introdução da norma 3.0, que serve de base a esta secção, nomeadamente com a introdução do HS³², foi possível atingir velocidades de 24 Mbit/s, alternando para uma ligação com base na norma 802.11, quando ambos os dispositivos são compatíveis, alternando do controlador EDR para um controlador AMP³³.

²⁵Infrared Data Association

²⁶Frequency-Hopping Spread Spectrum

²⁷Gaussian Frequency-Shift Keying

²⁸Enhanced Data Rate

²⁹Differential Quadrature Phase-Shift Keying

³⁰Differential Phase-Shift Keying

³¹Basic Rate

³²High Speed

³³Alternate PHY/MAC

Camada Física

O Bluetooth opera na banda ISM licenciada dos 2.4 GHz possuindo um total de 79 frequências espaçadas de 1 MHz. De modo a combater as interferências utiliza um mecanismo de espalhamento espectral por saltos em frequência FHSS, alternando assim entre várias frequências definidas previamente de modo pseudoaleatório.

A arquitetura de um sistema Bluetooth consiste sempre num elemento denominado por anfitrião (*Host*) e por um ou vários controladores (*Controllers*). O bloco anfitrião é constituído por todas as camadas abaixo dos perfis de utilização e acima da interface de HCI³⁴. Em contrapartida, o bloco controlador compreende todas as camadas abaixo da camada HCI.

Na versão 3.0 do Bluetooth existem dois tipos de controladores: BR/EDR, que inclui as camadas rádio, banda base, gestão de ligação (*link manager*) e HCI (opcionalmente), e AMP, que inclui AMP PAL³⁵, AMP MAC³⁶, PAL e, opcionalmente, a HCI.

O controlador BR/EDR já existia na versão anterior do Bluetooth, sendo que o controlador AMP foi a novidade introduzida nesta versão, acrescentando a função HS. Este controlador é secundário, e permite que, após dois dispositivos efetuarem uma conexão via BR/EDR, se for encontrado o controlador AMP no outro dispositivo, o tráfego de dados possa ser movido do controlador BR/EDR para o controlador AMP e os dados serem transferidos através de uma ligação 802.11, o que significa que o controlador AMP utiliza a camada física e MAC desta norma do IEEE para dar suporte à transmissão de grandes quantidades de tráfego. A Figura 2.13 apresenta as possíveis arquiteturas da versão 3.0 do Bluetooth, combinando o bloco anfitrião com o(s) bloco(s) controlador(es). Em a) existe apenas o controlador BR/EDR, em b) os controladores BR/EDR e AMP e em c) o controlador BR/EDR com múltiplos controladores AMP.

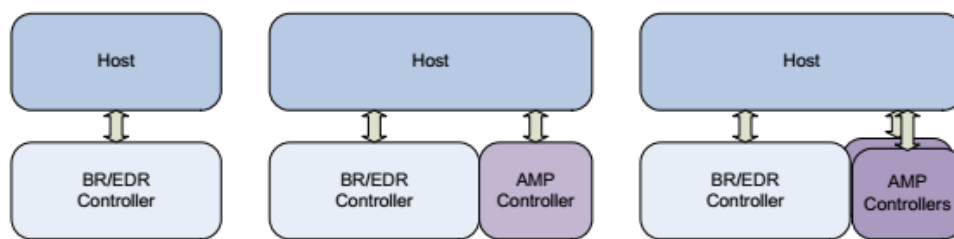


Figura 2.13: Arquitetura Bluetooth 3.0[32]

Em termos de débito, o Bluetooth possui um débito de 1 Mbps para a versão BR, recorrendo a uma modulação GFSK, 2 Mbps e 3 Mbps para a versão EDR, recorrendo às modulações $\frac{\pi}{4}$ -DQPSK e 8-DPSK, respectivamente, e um débito de 24 Mbps, caso seja usada a transmissão do tráfego via ligação IEEE 802.11.

³⁴Host Controller Interface

³⁵Protocol Adaptation Layer

³⁶Media Access Control

Camada MAC

A uma rede Bluetooth dá-se o nome de *piconet*, e nesta existem dois tipos de dispositivos: mestre e escravo. Um mestre inicia sempre as conexões, sendo o responsável pelo acesso ao meio dentro da *piconet*. Um dispositivo pode pertencer a mais do que uma *piconet*, formando assim uma *scatternet*. Numa *piconet*, os escravos associam-se ao mestre e a comunicação é feita sempre via mestre-escravo ou escravo-mestre através de um esquema de *polling*, não existindo por isso ligação direta entre escravos. A comunicação é gerida através de um esquema de divisão nos tempos, TDM³⁷. A Figura 2.14 apresenta um exemplo de uma comunicação entre um mestre e um escravo, em a), um mestre e vários escravos, em b), e de uma *scatternet* formada por três *piconets*, em c).

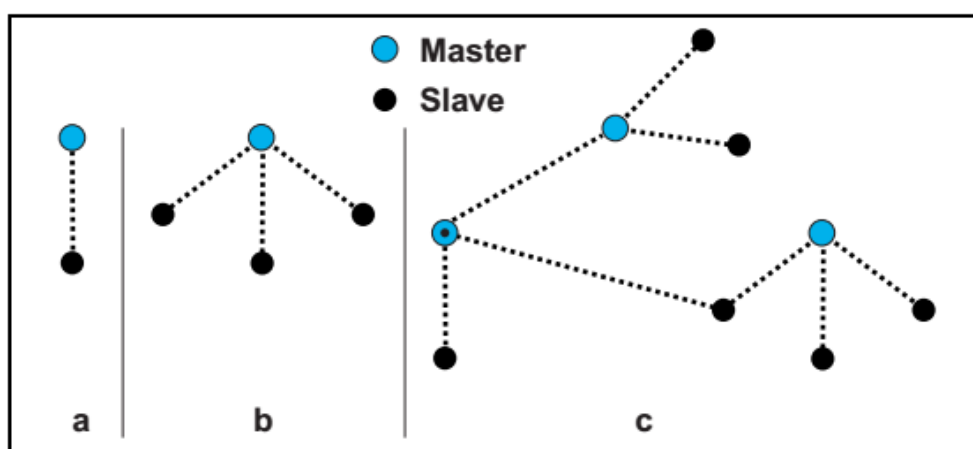


Figura 2.14: Exemplo de uma piconet com 1 escravo, multi-escravo e scatternet[32]

Numa piconet, a sequência de saltos é determinada pelo endereço do dispositivo mestre, composto por uma sequência única de 48 bits, o que origina para cada dispositivo mestre uma sequência de saltos também única. Ao ser garantido este passo, é pouco provável que duas redes que se encontrem próximas tenham saltos em comum na sua sequência, o que permite a coexistência dessas mesmas redes no mesmo espaço. Para que dois dispositivos possam comunicar entre si, necessitam de obter o endereço e sincronização com o relógio do mestre, para determinar a fase e a sequência de saltos da rede. Sem esta sincronização, os dispositivos não poderiam comunicar pois estariam desfasados entre si. Quando um dispositivo de uma piconet se associa a outra piconet obtém o estatuto de gateway, pois passa a fazer a ponte entre as duas piconets. Este *gateway* apenas pode ser mestre de uma piconet, pois não poderiam coexistir duas piconets com a mesma sequência de saltos. Além disso, o facto de as piconets possuírem sequências de salto diferentes, diminui a interferência causada pela proximidade das redes. Como foi referido, a interação entre mestre e escravo é feita através da divisão no tempo. Tal como sugere a Figura 2.15, o tempo está dividido em *slots* de 625 μs , e a duração da transmissão varia de acordo com o tamanho do pacote a enviar. Existem pacotes de ocupam 1,3 ou 5 *slots*, dependendo do seu tipo, definido neste standard.

³⁷Time Division Multiplexing

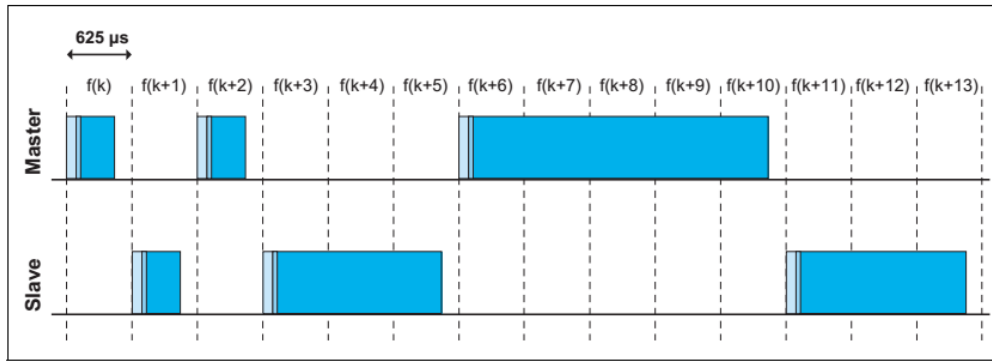


Figura 2.15: Comunicação entre mestre e escravo[32]

Através da imagem podemos observar que o mestre inicia sempre as suas transmissões em frequências pares, enquanto que um escravo inicia em frequências ímpares. Os tamanhos dos pacotes em *slots*, sendo números ímpares, fazem com que esta regra seja mantida entre mestre e escravo. Quando um pacote que utiliza mais do que um *slot* é transmitido, a frequência usada no início desse *slot* é mantida ao longo de toda a transmissão. No final da transmissão é efetuado o salto para a frequência correspondente ao próximo *slot*, saltando as frequências ocupadas pelo pacote.

Bluetooth 4.0

A versão 4.0 do standard Bluetooth, além de possuir os controladores para BR/EDR e AMP, possui ainda controladores para BLE, um sistema que foi criado com o objetivo de reduzir a quantidade de energia necessária para a transmissão de pequenas quantidades de dados, mantendo um custo reduzido, quando comparado com outros dispositivos que suportam BR/EDR. Com a definição da versão 4.0, foi possível obter arquiteturas de interação entre dispositivos que suportam diferentes controladores, como ilustrado na figura 2.16, podendo ser possível escolher entre disponibilizar ou não cada um dos controladores primários (BLE e BR/EDR) e conter, ou não, um ou mais controladores secundários (AMP).

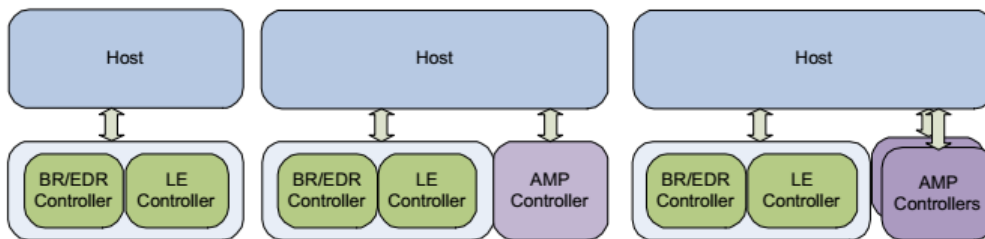


Figura 2.16: Arquitetura Bluetooth 4.0[32]

Os dispositivos que suportam ambos os controladores primários, BLE e BR/EDR, são designados por *dual-mode* e pertencem à categoria *Bluetooth Smart Ready*, categoria na qual se inserem os *smartphones* e os computadores portáteis, que suportem *Bluetooth 4.0*. Os dispositivos que suportam apenas o controlador BLE são referidos como sendo dispositivos

single-mode e inserem-se na categoria *Bluetooth Smart*, na qual se insere a maioria dos sensores existentes que sejam alimentados por pilhas e baterias. Os dispositivos que apenas suportam os controladores BR/EDR são inseridos na categoria *Bluetooth*. A figura 2.17 representa a interação entre as diferentes arquiteturas, demonstrando a interligação entre dispositivos com apenas um dos controladores com os dispositivos que suportam ambas as tecnologias.

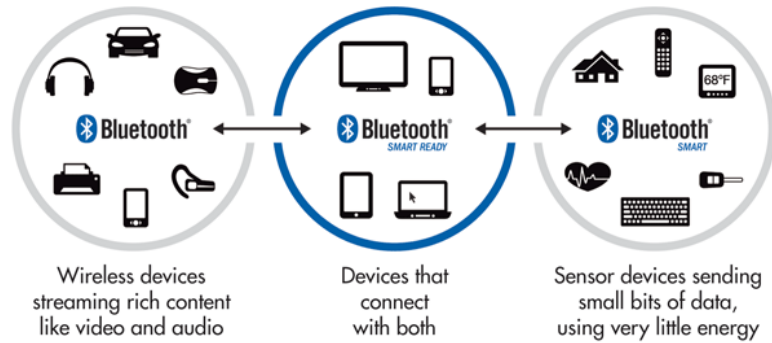


Figura 2.17: Interação entre sistemas Bluetooth[33]

2.2.4 Bluetooth Low Energy

A especificação da versão 4.0 pelo Bluetooth SIG define-a como sendo uma tecnologia de comunicação sem fios de curto alcance, de muito baixo consumo energético, ULP³⁸, com uma pilha protocolar leve e que permite a integração com a tecnologia Bluetooth existente. Em comparação com as versões que existiam antes, pode-se observar na figura 2.18 que foi adicionada a camada de *profile* às camadas *Host* e *Controller* já existentes, o que revela que todos os perfis e aplicações se situam no topo da pilha protocolar.

³⁸Ultra Low Power

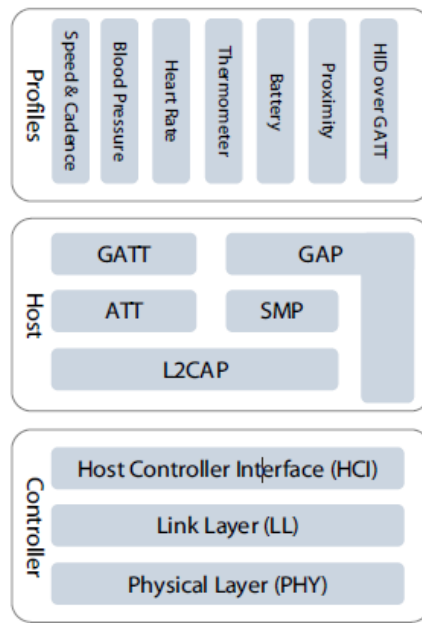


Figura 2.18: Pilha Protocolar BLE[32]

Camada Física

O BLE opera na banda de frequências de 2,4 GHz e define 40 canais de rádio frequência. Existem dois tipos de canais RF³⁹ no BLE: canais de anúncio e canais de dados. Os canais de anúncio são três e são usados para descobrir dispositivos, estabelecer conexões e para transmissões *broadcast*, enquanto que os 37 canais restantes de dados são usados para comunicação bidirecional entre os dispositivos conectados. É usado o mecanismo de espalhamento espectral por saltos em frequência, FHSS, nos canais de dados para evitar interferências e problemas de propagação no meio sem fios, como o desvanecimento (*fading*) ou as reflexões por múltiplos caminhos (*multipath*). Este mecanismo seleciona um dos 37 canais disponíveis para comunicação durante um dado intervalo de tempo. O facto de existirem apenas três canais de anúncio torna mais provável a ocorrência da escolha do mesmo canal por parte de outra tecnologia de transmissão, resultando na degradação do sinal, mas, para combater esse facto, os canais de anúncio do BLE foram cuidadosamente escolhidos de modo a não coincidirem com os canais standard do IEEE 802.11. A Figura 2.19 ilustra o mapeamento das frequências BLE para os canais de dados e de anúncio.

³⁹Radio-Frequência

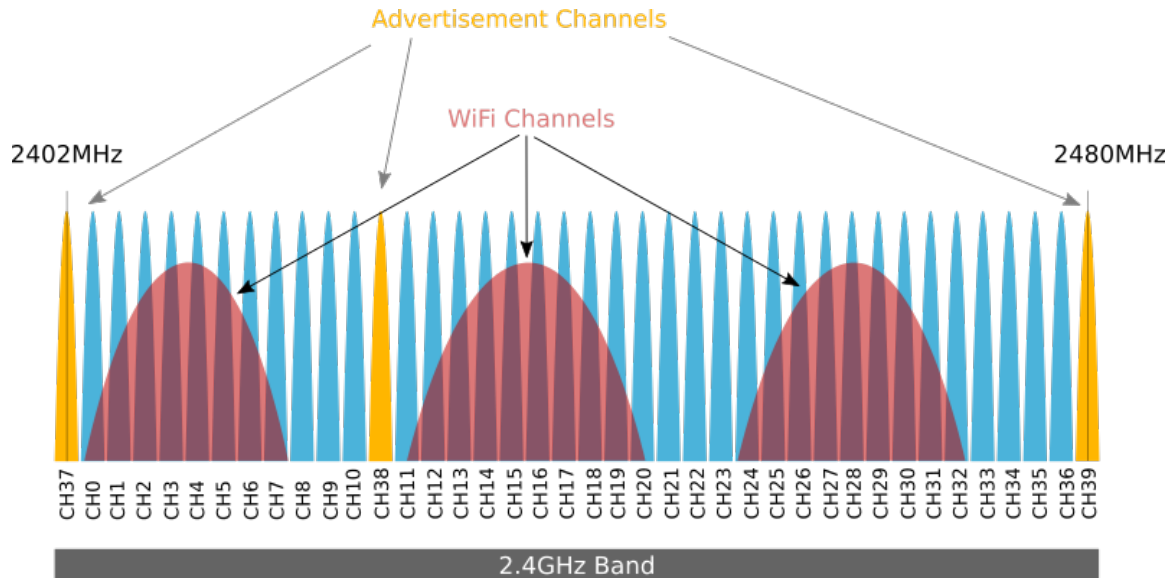


Figura 2.19: Canais de comunicação BLE[34]

Todos os canais da camada física usam modulação Gaussiana, GFSK, que é baseada na modulação FSK⁴⁰, onde a diferença reside na introdução de um filtro gaussiano para diminuir a largura espectral dos impulsos suavizando a transição entre os valores dos mesmos, permitindo também reduzir o pico de energia consumida. A taxa de transmissão na camada física é de 1 Mbps e o seu alcance é de aproximadamente 150 metros em zona ampla, livre de obstruções. A sensibilidade do recetor, segundo a especificação, é definida como o nível de sinal necessário para que a taxa de erro por bit, BER⁴¹ seja de 10^{-3} . Esse valor deve ser igual ou melhor que -70 dBm, o que garante vários metros de cobertura.

Camada Ligação

A operação da camada de ligação pode ser descrita em termos de uma máquina de estados que opera em 5 estados distintos, sendo eles: standby, anúncio (advertising), descoberta (scanning), iniciação (initiating) e conexão (connection), tal como sugere a Figura 2.20. Apenas um destes estados está activo num certo período de tempo.

No estado de *standby* não são transmitidos ou recebidos quaisquer pacotes, e este estado pode ser acedido a partir de qualquer outro estado. No estado de anúncio é possível transmitir pacotes de anúncio para a rede e há a possibilidade de ouvir e responder a respostas desencadeadas por esses canais de anúncio. Um dispositivo neste estado é chamado de anunciante (advertiser) e este estado pode ser acedido através do estado de *standby*. A camada de ligação a operar no estado de descoberta escuta os canais de anúncio procurando por pacotes de anúncio vindos de dispositivos anunciantes. Um dispositivo neste estado é chamado de *scanner*, e o estado anterior a este é sempre o de *standby*.

No estado de iniciação ocorre a escuta por pacotes de anúncio vindos de dispositivos específicos, e ocorre a resposta a esses canais de forma a iniciar-se uma ligação com esse

⁴⁰Frequency Shift Keying

⁴¹Bit Error Rate

dispositivo. Um dispositivo que inicia uma conexão é chamado de iniciador, e transita para este estado vindo do estado de *standby*. Por fim, o estado de conexão pode ser acedido pelo estado de anúncio ou de iniciação, pois um dos dispositivos estava a anunciar e o outro é que descobriu e iniciou a ligação. Assim, aquando num estado de conexão, dois papéis são definidos ao nível da camada de ligação: mestre e escravo. A um dispositivo que chega ao estado de conexão vindo de um estado de iniciação, é-lhe atribuído o papel de mestre, assim como ao dispositivo que transitou do estado de anúncio para o estado de conexão é atribuído o papel de escravo. A camada de ligação no papel de mestre comunica com um ou mais dispositivos escravos e define os tempos de transmissão dos dados. Já no papel de escravo, a camada de ligação comunica apenas com um dispositivo mestre.

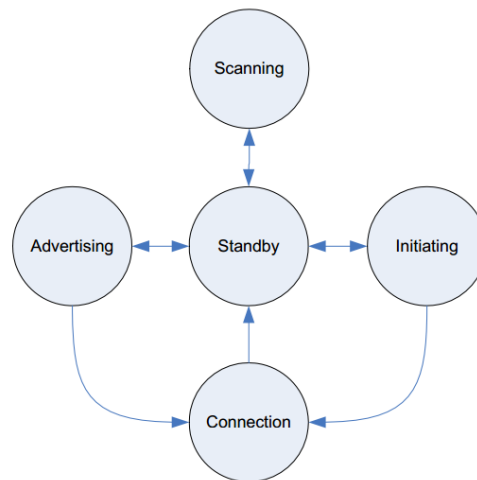


Figura 2.20: Máquina de Estados BLE[35]

Existem 3 parâmetros essenciais ao funcionamento de todo o protocolo MAC do BLE, que são definidos no momento em que é efetuado o estabelecimento de uma conexão, nomeadamente o intervalo de conexão, a latência do escravo e o *timeout* de supervisão. Estes parâmetros podem ser descritos como:

- Intervalo de ligação (*connection interval*): Numa ligação BLE entre dois dispositivos é usado o esquema de saltos em frequência de modo a garantir que os dois troquem informação num dado canal num certo instante de tempo. Este momento é chamado de evento de ligação (*connection event*). Assim sendo, o intervalo de ligação é o período de tempo que decorre entre dois intervalos de conexão, sendo que o valor mínimo corresponde a 7,5 milissegundos e o valor máximo é de 4 segundos. Consoante o tipo de aplicação, podem ser necessários diferentes intervalos de conexão;
- Latência do escravo (*slave latency*): Este parâmetro confere ao dispositivo periférico (escravo) a opção de saltar um certo número de eventos de conexão. Garante mais flexibilidade ao dispositivo periférico, dado que, se não tiver dados para enviar, pode escolher saltar os próximos eventos de conexão e ficar a dormir, sem ter de acordar para escutar o mestre, contribuindo assim para uma poupança ao nível energético. O

valor atribuído a esta latência representa o número máximo de eventos que podem ser ultrapassados podendo ir de um mínimo de 0 até ao máximo de 499;

- Timeout de supervisão (supervision timeout): Uma ligação pode falhar devido a vários motivos, como o caso de um dispositivo se mover para fora da área de alcance, zonas de elevada interferência ou até falha de bateria por parte de um dos dispositivos. Posto isto, é importante que tanto o mestre como o escravo tenham controlo sobre o estado de ligação e, por isso, aquando do pedido de conexão, é definido o valor de *timeout* de supervisão, sendo que este é reiniciado a cada vez que se envia um novo pacote. Desta forma, se após a transmissão este valor for excedido, a ligação é considerada perdida.

Interface Controlador-Anfitrião (HCI)

A camada HCI fornece os meios de comunicação entre a secção controladora (*Controller*) e a secção anfitriã (*Host*) através de interfaces padronizadas. Esta camada pode ser implementada quer através de uma API⁴², quer por interfaces de hardware como UART⁴³, SPI⁴⁴ ou USB⁴⁵.

Protocolo de Controlo e Adaptação da Ligação Lógica (L2CAP)

O L2CAP⁴⁶ gere o encapsulamento dos dados para as camadas superiores, permitindo gerar uma comunicação lógica virtual entre dois dispositivos, criando a abstracção ao *hardware* utilizado. Este processo é feito numa abordagem de melhor esforço *best-effort*, não utilizando mecanismos de retransmissão nem controlo de fluxo, ao contrário da metodologia utilizada noutras versões do *Bluetooth*. Os protocolos utilizados pelas camadas superiores produzem estruturas de dados que cumprem com o tamanho máximo dos dados do L2CAP, que corresponde a 23 *bytes*.

Camada de Gestão da Segurança

A camada de gestão de segurança define os mecanismos para emparelhamento e distribuição de chaves e fornece funções para que outras camadas da pilha se liguem de forma segura e troquem dados com outros dispositivos. Os dados são lidos ou escritos apenas quando a ligação é autenticada, requerendo que, no momento em que esta for estabelecida, ambos os dispositivos passem por um processo denominado de emparelhamento (*pairing*). Quando este processo ocorre, as chaves são definidas e utilizadas quer para encriptar quer para autenticar a ligação. Nos casos normais, o dispositivo periférico irá requisitar uma chave ao dispositivo central de modo a completar o processo de emparelhamento. Este pode ser um valor fixo, como “000000”, ou outro fornecido pelo utilizador. Depois do dispositivo central enviar corretamente esta chave, os dois dispositivos trocam as chaves de segurança para encriptar e autenticar a ligação. Em muitos casos, o dispositivo central e o periférico estabelecem e quebram a ligação

⁴²Application Programming Interface

⁴³Universal Asynchronous Receiver/Transmitter

⁴⁴Serial Peripheral Interface

⁴⁵Universal Serial Bus

⁴⁶Logical Link Control and Adaptation Protocol

diversas vezes. Para facilitar este processo, o BLE tem uma função de segurança denominada por *bonding* que permite aos dois dispositivos, no processo de emparelhamento, trocar um conjunto de chaves de segurança, com uma validade mais extensa. Esta função permite que dois dispositivos restabeleçam rapidamente a encriptação e autenticação após voltarem a estabelecer nova ligação, sem terem de passar novamente pelo processo de emparelhamento. A camada de ligação suporta encriptação e autenticação usando um algoritmos de cifras por blocos. Quando se aplica a encriptação e autenticação a uma conexão, são acrescentados 4 bytes ao payload do canal, chamados *Message Integrity Check* (MIC). Também é possível transmitir dados autenticados através de uma ligação não encriptada, pelo que, neste caso, uma assinatura de 12 *bytes* é colocada depois do *payload* na camada ATT⁴⁷. A assinatura é feita aplicando um algoritmo que usa AES⁴⁸ de 128 *bits* como cifra de blocos.

ATT (Attribute Protocol)

Este protocolo permite expor a outros dispositivos certos componentes dos seus dados, chamados de atributos. Os atributos são estruturas de dados que armazenam informações fornecidas pela camada GATT⁴⁹, que será abordada no próximo ponto. Um atributo é composto por 4 elementos: 16 bits para o *handle* que define unicamente um atributo, um UUID⁵⁰ de 16 ou 32 bits que define o tipo de atributo, a sua descrição e por fim o seu valor de tamanho variável.

No contexto do ATT, um dispositivo que expõe os seus atributos é referido como o servidor e o dispositivo vizinho é o cliente. Os papéis de mestre/escravo atribuídos na camada de ligação são independentes dos papéis de cliente/servidor definidos neste protocolo, o que permite que um dispositivo mestre tanto possa ser um cliente e/ou um servidor, acontecendo o mesmo com o dispositivo que toma o papel de escravo, sendo ainda possível um dispositivo ser cliente e servidor em simultâneo. O cliente pode aceder aos atributos do servidor através do envio de pedidos, e o servidor, de modo a garantir eficiência, pode enviar ao cliente dois tipos de mensagens não solicitadas contendo os atributos, nomeadamente: notificações, se o servidor enviar dados ao cliente mas este não tiver que confirmar a sua receção, ou indicações, se o envio exigir uma confirmação por parte do cliente.

GATT (Generic Attribute Profile)

A camada GATT define uma *framework* para usar os procedimentos da camada ATT para descoberta de serviços e especifica a estrutura dos perfis. Todas as comunicações de dados que ocorrem entre dispositivos numa conexão BLE são processadas através de procedimentos desta camada, em que todos os dados que estão a ser usados para um serviço são chamados de características. Uma característica é, por sua vez, um conjunto de dados que inclui um valor e certas propriedades. Os dados relativos ao serviço e às suas características são guardados em atributos. De forma hierárquica podemos dizer que um perfil pode ter vários serviços e

⁴⁷Attribute Protocol

⁴⁸Advanced Encryption Standard

⁴⁹Generic Attribute

⁵⁰Universally Unique Identifier

que cada serviço pode ter várias características. A definição dos serviços e características está organizada sob a forma de atributos, mapeados numa tabela chamada de tabela de atributos. Como já foi referido, toda esta informação está definida na camada GATT, mais propriamente no dispositivo servidor, pois é neste que estão definidos os vários serviços. O dispositivo cliente tem a possibilidade de aceder a estes atributos através de operações de leitura e escrita, consoante as permissões implementadas. Segundo as normas definidas pelo Bluetooth SIG, a definição de um serviço tem como UUID o valor 0x2800[32]. Todos os atributos seguintes pertencem a este serviço até ser encontrado novamente o mesmo UUID, significando assim o início de um novo serviço, como se observa na tabela 2.3.

Handle	UUID	Descrição	Valor
0x0100	0x2800	Serviço 1	UUID = 0x1802
...	...	Detalhes do serviço 1	...
0x0150	0x2800	Serviço 2	UUID = 0x188F
...	...	Detalhes do serviço 2	...

Tabela 2.3: Exemplo de definição de dois serviços

As características, como parte integrante dos serviços, são identificadas com um UUID próprio, que corresponde a 0x2803. Este identificador delimita uma característica, sendo que esta, obrigatoriamente, tem no mínimo dois atributos, um para a identificar e outro para definir o seu valor. Se necessário, uma característica pode ter mais atributos extra, que são designados por descritores e servem para dar mais informação acerca do valor da característica, nomeadamente as suas unidades ou grandeza medida, por exemplo, como se pode observar na tabela 2.4.

Handle	UUID	Descrição	Valor
0x0100	0x2800	Serviço 1	UUID = 0x1802
0x0101	0x2803	Característica 1	UUID = 0x2A2B; Handle = 0x0102
0x0102	0x2A2B	Valor Característica 1	Ex. 1021
0x0104	0x2A2F	Descritor 1	milibar
0x0110	0x2803	Característica 2	UUID = 0x2A2D; Handle = 0x0111
0x0111	0x2A2D	Valor Característica 1	Ex. 3.5

Tabela 2.4: Exemplo de definição de duas características para um serviço

Tal como acontece nos serviços, sabe-se que este termina onde for encontrado um novo UUID de definição de serviço (0x2800). O mesmo acontece com as características, ou seja, uma característica acaba e a próxima inicia-se sempre que na tabela de atributos se encontrar um UUID igual a 0x2803. Este valor é também um valor padrão da norma definida pelo Bluetooth SIG. Existem vários descritores já definidos pela norma e um dos mais importantes

é o descritor de configuração da característica do cliente, CCC⁵¹. Este descritor, cujo UUID é 0x2902, é um *bitmap* de 16 bits que podem ser lidos e escritos, sendo um atributo como qualquer outro do lado do servidor, sendo que os seus dois primeiros bits já estão tomados pela especificação do GATT, servindo para configurar se o valor da característica ao qual está associado será enviado para o cliente sob a forma de notificação ou indicação. O bit **0** informa sobre notificações e o bit **1** sobre indicações. A incorporação deste descritor é apresentada na Tabela 2.5, na qual se assume que as notificações estão ativadas.

Handle	UUID	Descrição	Valor
0x0100	0x2800	Serviço 1	UUID = 0x1802
0x0101	0x2803	Característica 1	UUID = 0x2A2B; Handle = 0x0102
0x0102	0x2A2B	Valor Característica 1	Ex. 1021
0x0104	0x2A2F	Descritor 1	milibar
0x0105	0x2902	Descritor CCC	0x0001
0x0110	0x2803	Característica 2	UUID = 0x2A2D; Handle = 0x0111
0x0111	0x2A2D	Valor Característica 1	Ex. 3.5

Tabela 2.5: Incorporação do descritor CCC no exemplo do serviço 1

GAP (Generic Access Profile)

No topo da pilha protocolar do BLE, encontra-se a camada GAP⁵². Esta camada é a interface direta com as aplicações e/ou perfis, sendo também responsável pela especificação dos papéis dos dispositivos, modos para a descoberta de dispositivos e serviços, gestão do estabelecimento e término de ligação, assim como a iniciação do processo de segurança. Esta camada opera sempre num dos seguintes 4 papéis: anunciador, observador, periférico ou central.

Um dispositivo no papel de anunciador apenas difunde informação nos canais de anúncio, não suportando ligações a outros dispositivos. Um dispositivo observador é o inverso do anunciador, pelo que apenas escuta anúncios, não tendo capacidade de se ligar. Em complemento a estes, um dispositivo central procura por anúncios e inicia ligações, funcionando como mestre e sendo capaz de gerir várias ligações. Um dispositivo periférico é um anunciador que tem a capacidade de se ligar, atuando como escravo numa ligação simples, ou seja, apenas se liga a um mestre.

Num sistema BLE típico, um dispositivo anuncia que se encontra pronto para estabelecer uma ligação, dando conhecimento aos dispositivos centrais. Estes anúncios contêm o endereço único do dispositivo e podem conter dados adicionais, como o nome do dispositivo. Na figura 2.21, o dispositivo central, após receber o anúncio, envia ao dispositivo periférico um *scan request*, ao qual este lhe responde com um *scan response*. Assim se encontra descrito o processo de descoberta dos dispositivos, na medida em que o dispositivo central passa a conhecer o

⁵¹Client Characteristic Configuration

⁵²Generic Access Profile

dispositivo com quem comunicou, reconhecendo que se trata de um dispositivo periférico, que pode estabelecer uma ligação. Após este passo, o dispositivo central envia um pedido de estabelecimento de ligação, *link request* ao dispositivo periférico, o qual responde com um *link request*.

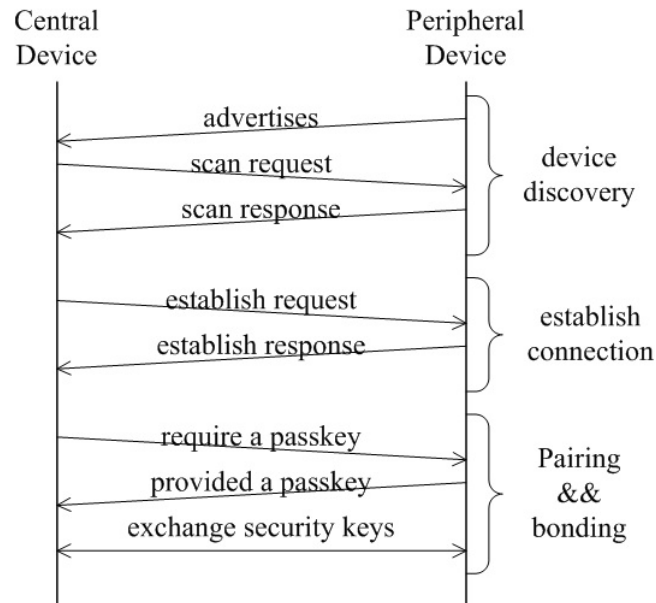


Figura 2.21: Sequência de ligação mestre-escravo BLE[36]

Formato dos Pacotes

O controlador LE⁵³ inserido nesta nova versão do Bluetooth apresenta apenas um formato de pacotes, quer para os que circulam nos canais de anúncios, quer para os que circulam nos canais de dados. O pacote consiste em 4 campos distintos: preâmbulo (1 byte), endereço de acesso (4 bytes), PDU⁵⁴ (2 a 39 bytes) e CRC⁵⁵ (3 bytes), tal como se pode observar na Figura 2.22.

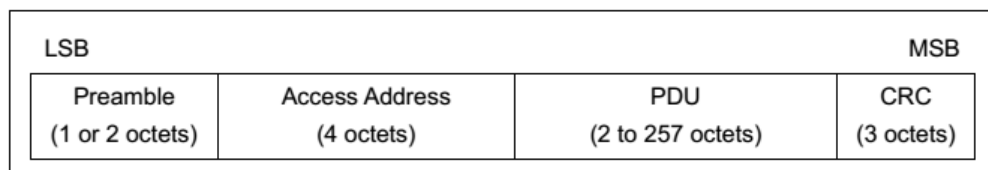


Figura 2.22: Formato geral de um pacote BLE[32]

O tamanho de um pacote BLE pode variar de 10 bytes (80 bits) até 47 bytes (376 bits), a maior dimensão que este pode ter. O preâmbulo é usado no recetor para executar a sincronização da frequência, estimar o sincronismo de um símbolo e para o controlo do ganho automático, enquanto que o endereço de acesso é um número de 32 bits, único e gerado

⁵³Low Energy

⁵⁴Protocol Data Unit

⁵⁵Cyclic Redundancy Check

aleatoriamente pelo dispositivo no estado de iniciação e enviado no pedido de conexão. O CRC é um código polinomial de 24 bits, que serve de mecanismo de identificação de erros e é calculado sobre o PDU do pacote, sendo que todos os bits do PDU devem ser processados pela ordem de transmissão, ou seja, do menos para o mais significativo. Relativamente ao PDU, existem dois tipos, dependendo se o pacote é transmitido nos canais de anúncios ou nos canais de dados.

PDU dos Canais de Anúncio

O PDU que corresponde aos canais de anúncio, visível na Figura 2.23, é constituído por um cabeçalho (*header*) de 2 bytes, visível na Figura 2.24, seguido do respetivo *payload*, que é específico para este tipo de PDU.

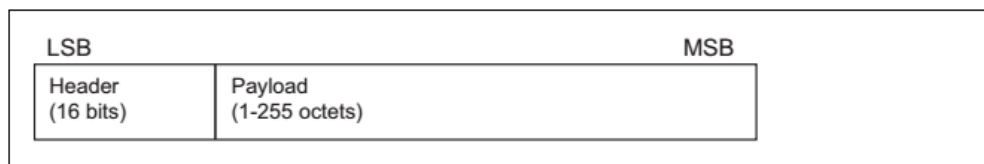


Figura 2.23: Formato do PDU no canal de anúncio[32]

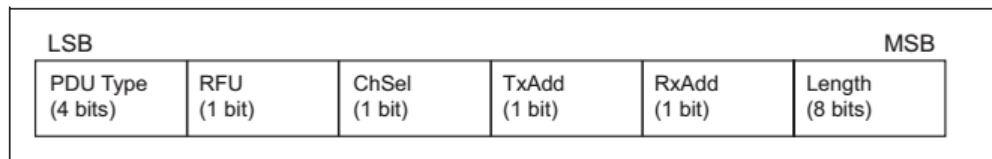


Figura 2.24: Formato do Header do PDU no canal de anúncio[32]

O campo *header* é composto por 6 subcampos. O *PDU Type* indica qual o tipo de PDU, dos subtipos definidos para esta categoria, descritos na tabela 2.6. Os campos ***TxAdd*** e ***RxAdd*** tomam certos valores (**0** ou **1**), dependendo do tipo de PDU. Caso não sejam utilizados, são reservados para uso futuro, RFU⁵⁶. O campo ***length*** é o tamanho do payload do PDU em bytes, que como está descrito na figura pode ir desde 6 a 37 bytes.

⁵⁶Reserved for Future Use

Tipo de PDU	Nome do Pacote
0000	ADV_IND
0001	ADV_DIRECT_IND
0010	ADV_NONCONN_IND
0011	SCAN_REQ
0100	SCAN_RSP
0101	CONNECT_REQ
0110	ADV_SCAN_IND

Tabela 2.6: Subtipos de PDU do canal de anúncios

PDU dos Canais de Dados

O PDU do canal de dados é constituído por um *header* (2 bytes), um *payload* de tamanho variável (0 a 27 bytes) e um campo MIC⁵⁷ opcional (4 bytes). O MIC não está incluído nas situações em que se verifica uma conexão não encriptada ou uma conexão encriptada com o payload do PDU igual a zero. A Figura 2.25 ilustra o formato do PDU do canal de dados, correspondendo ao pacote genérico, que aceita virtualmente um payload de 256 bytes, ainda que o seu tamanho máximo corresponda aos 27 bytes anteriormente referidos.

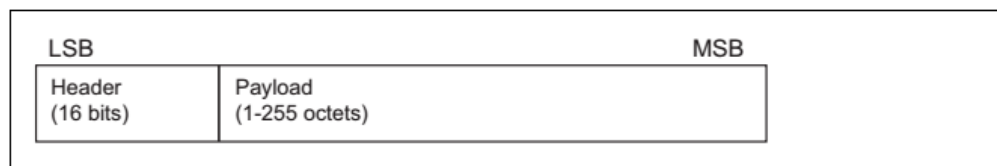


Figura 2.25: Formato do PDU no canal de dados[32]

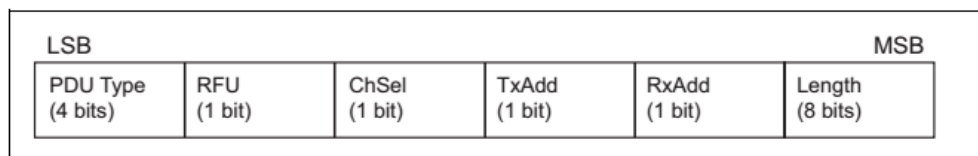


Figura 2.26: Formato do Header do PDU no canal de dados[32]

O campo **LLID** define o tipo de *payload*. Se tiver o valor 01b (1) ou 10b (2), contém um *payload* de dados, do tipo **LL DATA PDU**, caso tenha o valor 11b (3), então o *payload* é de controlo, do tipo **LL CONTROL PDU**. Um *payload* de dados significa que irá conter dados vindos da camada L2CAP, e se o valor for 01b significa que o *payload* é vazio. Este tipo de PDU é enviado pela camada de ligação no papel de mestre para permitir ao escravo responder com qualquer PDU de dados, incluindo um PDU vazio. Caso o valor seja 10b, o payload irá conter dados, ou seja, o seu tamanho é diferente de zero. Os PDUs de controlo têm como

⁵⁷Message Integrity Check

função garantir o controlo e manutenção da conexão ao nível da camada de ligação. O campo **NESN** indica o próximo número de sequência esperado, ou seja, é usado pelo recetor para reconhecer o último PDU enviado ou para requisitar um novo envio do mesmo. O campo **SN** contém o número de sequência usado para identificar os pacotes enviados pela camada de ligação.

O campo **MD** significa *more data*, e permite indicar se o dispositivo tem mais dados para enviar. Se nenhum dispositivo tem o bit MD ativo nos seus pacotes, o pacote enviado a partir do escravo fecha a ligação. Caso um ou ambos os dispositivos tenham o bit MD ativo, o mestre continua com o evento de conexão enviando um novo pacote, devendo o escravo escutar depois de enviar o seu pacote. Se um pacote não for recebido pelo mestre vindo do escravo, o mestre termina o evento de conexão. Caso um pacote seja enviado pelo mestre e o escravo não receba, este último termina o evento de conexão. Por fim, o campo **length**, de 5 bits, indica o tamanho do payload e do MIC, sendo por isso o seu tamanho máximo limitado a 31 bytes. Como o MIC tem um tamanho fixo de 4 bytes, o payload deve ser igual ou menor que 27 bytes. Como foi referido anteriormente, caso o payload do PDU não seja vazio, irá conter os dados vindo da camada L2CAP. O formato dos dados nesta camada reserva 4 bytes para o cabeçalho e os restantes 23 bytes para dados. Por sua vez o payload da camada L2CAP define um byte para o *opcode*, restando 22 bytes para dados, sendo que 2 desses bytes então destinados à indicação da **ATT_HANDLE** ou seja a posição de memória onde se encontra o atributo que contém os dados a serem enviados. Posto isto conclui-se que o tamanho máximo de dados de utilizador disponível para envio num pacote BLE é de 20 bytes. A Figura 2.27 mostra o formato do payload do PDU do canal de dados, enquanto que a figura 2.28 mostra o formato do cabeçalho do pacote L2CAP

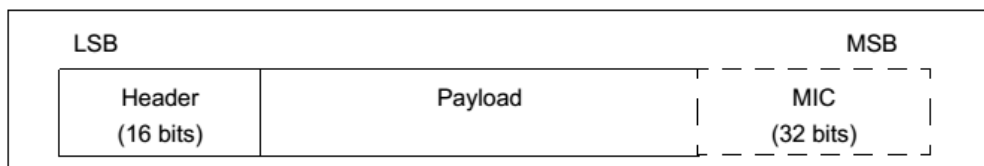


Figura 2.27: Formato do *payload* do PDU de dados[32]

Header					
LLID (2 bits)	NESN (1 bit)	SN (1 bit)	MD (1 bit)	RFU (3 bits)	Length (8 bits)

Figura 2.28: Formato do cabeçalho do pacote L2CAP[32]

Arquitetura do Sistema Inicial

Surgindo em 2010, o sistema Bikeemotion foi evoluindo progressivamente desde uma prova de conceito até ser um sistema comercial, como atualmente. Ao longo do tempo, várias tipologias de desenvolvimento foram experimentadas, passando-se por uma fase de protótipo para bicicletas tradicionais, com um *hardware* genérico e aproveitado de outros projetos da empresa, até à solução atual, com *hardware* dedicado desenhado à medida, sendo compatível tanto com sistemas de bicicletas tradicionais como com sistemas de bicicletas elétricas.

Este capítulo consiste numa apresentação inicial da origem e evolução do sistema Bikeemotion, seguida da descrição detalhada do sistema na sua totalidade, nomeadamente no que se refere ao *hardware* utilizado, destaque para a bicicleta, e ao modelo de funcionamento do sistema.

3.1 ORIGEM E EVOLUÇÃO DO PROJETO

O projeto original foi concebido com base num protótipo desenvolvido recorrendo a placas já existentes, ilustradas na figura 3.1, como prova de conceito. Esta solução protótipo era constituída por uma placa de circuito impresso Unicard Anywhere desenvolvida pela Micro I/O e por uma aplicação desenvolvida em Java para sistemas embutidos, sendo executada na máquina virtual Java existente no modem TC65i da Siemens. Associado a estes componentes, existia ainda um leitor de RFID SL031 da StrongLink, responsável por fazer a leitura dos cartões, estando este ligado por uma interface UART ao modem. Para ser possível obter a localização da bicicleta, recorreu-se ao um localizador GPS fornecido pela uBlox.

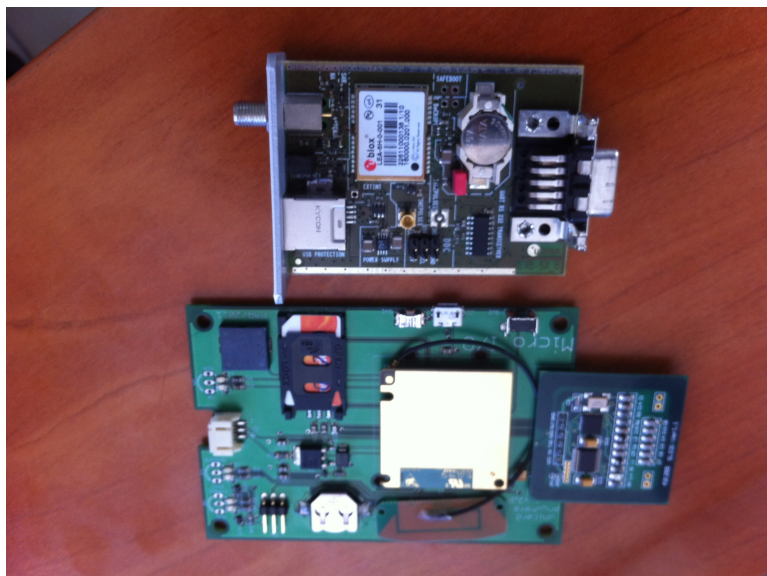


Figura 3.1: Sistema inicial Bikeemotion

Nesta versão, o sistema funcionava de modo análogo ao da prisão domiciliária, em que a guarda física é trocada por um elemento de vigilância eletrônica. Dessa forma, é definido um parque virtual georreferenciado, que define os limites que, ao existir uma bicicleta que abandone essa área sem ter sido previamente desbloqueada, irá dar origem a uma notificação e a posição da mesma será enviada com uma frequência fixa, permitindo a localização da mesma no terreno.

As bicicletas são equipadas com um leitor de RFID que permite que sejam desbloqueadas com a passagem de um cartão. Ao existir a georreferenciação dos pedidos de bloqueio e desbloqueio das bicicletas, apenas será permitido efetuar uma destas operações caso a bicicleta se encontre dentro da área de um parque válido, fora do qual este pedido será negado. Ao não ser devolvida a bicicleta na área segura definida pelo fornecedor do serviço, as bicicletas não podem ser desassociadas do utilizador, pelo que continuam à responsabilidade do mesmo, até serem convenientemente devolvidas.

3.2 HARDWARE

No sistema Bikeemotion, todo o *hardware* do sistema está contido num conjunto de duas placas de circuito impresso que são unidas numa montagem em *sandwich*, o que permite que todo esse equipamento consiga ser alojado no espaço existente no compartimento existente no guiador da bicicleta. Estas duas placas, denominadas por placa-mãe e placa de comunicações, cujas fotografias são apresentadas nas figuras 3.3 e 3.4, respetivamente, são as unidades de controlo responsáveis pela leitura de sensores e atuação do motor da bicicleta, assim como a gestão de energia e carregamento das baterias, como representado pelo diagrama de blocos apresentado na figura 3.2, que representa a visão geral do sistema de *hardware*, realçando inequivocamente cada um dos subsistemas considerados.

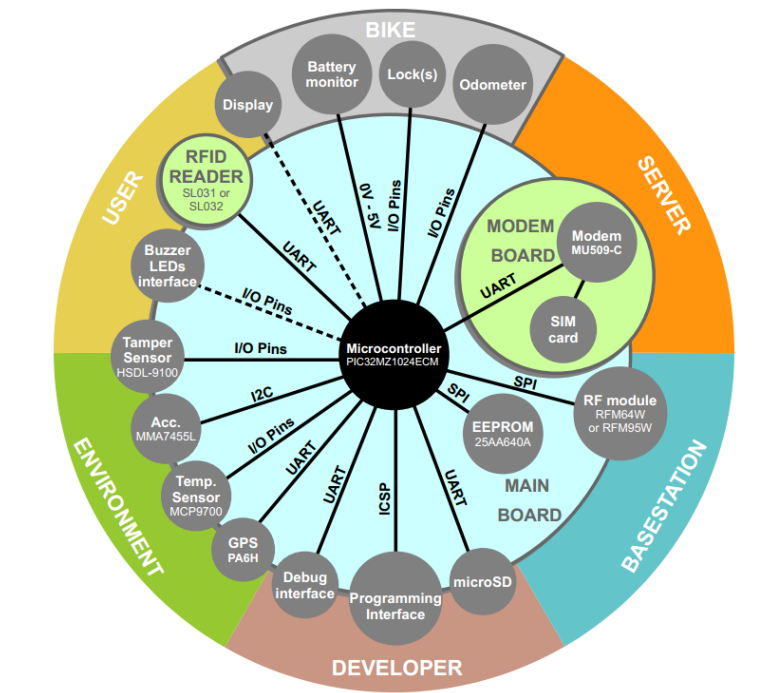


Figura 3.2: Diagrama de blocos do projeto Bikeemotion

Ao abranger todas as áreas de interação representadas no diagrama de blocos, o *hardware* das bicicletas é autónomo, podendo funcionar de forma desligada dos servidores em caso de necessidade.

3.2.1 Placa-mãe

A placa-mãe, ilustrada pela figura 3.3, é o elemento do sistema que aloja os componentes de processamento de dados do sistema, os módulos de GPS, acelerómetro e giroscópio, interface com o cartão microSD¹, sensor de temperatura e sensor de vandalismo.

O microcontrolador é o elemento central do sistema, executando todo o processamento de dados dos sensores existentes. O componente escolhido para a solução foi o PIC32MZ do fabricante Microchip, tendo sido escolhido este modelo em particular devido à necessidade de valores de memória de programa e de RAM² bastante superiores aos valores encontrados nas restantes opções existentes no mercado. O modelo escolhido tem como valores de memória 1MB para código de execução e 512KB para memória RAM. Em execução, o código executável existente ocupa mais de 700KB da memória, ao mesmo tempo que ocupa mais de 400KB de memória RAM.

¹Secure Digital

²Random Access Memory

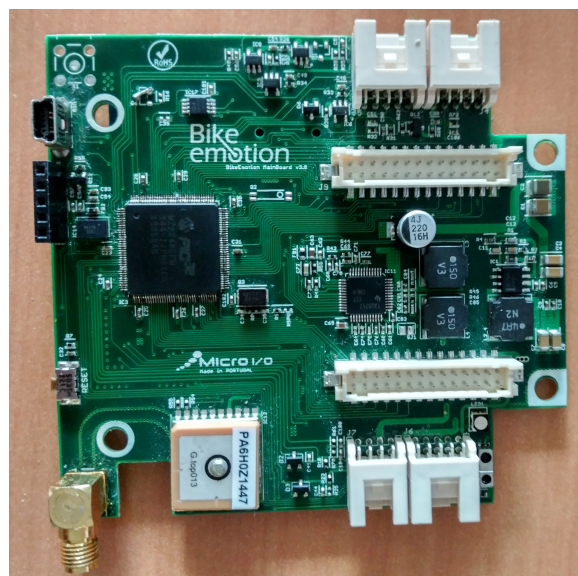


Figura 3.3: Placa-mãe

O módulo de GPS permite a localização geográfica das bicicletas em tempo real, recorrendo à localização por satélite. O componente escolhido para desempenhar esta função no sistema foi o PA6H do fabricante GlobalTop, também conhecido pela designação *LadyBird*. A localização obtida por este módulo é utilizada para o registo as rotas efetuadas pelos utilizadores e para localização das bicicletas no terreno, principalmente em situações de roubos ou furtos. Este módulo comunica com o microcontrolador através de uma interface UART, dispondo de uma ligação para a utilização de uma antena externa, que se encontra implementada através de um conetor SMA³.

Os sensores de acelerómetro e giroscópio permitem obter a aceleração das bicicletas, assim como obter a orientação e inclinação das mesmas. O componente escolhido para este sistema é o LIS2DE da STMicroelectronics, que consiste num elemento capaz de detetar variações nos três eixos, com alcances de $\pm 16G$, em que G corresponde à aceleração da gravidade ao nível médio da água do mar. Este módulo comunica através de uma interface I2C⁴, contendo um processador de movimento integrado, capaz de detetar eventos de queda livre, variação de velocidade num dado sentido e notificação de ultrapassagem de patamares definidos pelo programador. Este elemento é utilizado para a deteção de vandalismo e acidentes durante a utilização da bicicleta.

O encaixe para cartões microSD permite a inserção na placa-mãe de um cartão desta família. Este dispositivo comunica através de uma interface SPI, permitindo o armazenamento de conteúdos gráficos, configurações e registos de execução numa unidade de armazenamento amovível. Todos os conteúdos utilizados para a interface gráfica do ecrã LCD ou interface com o utilizador estão armazenados no cartão, tais como imagens ou ficheiros de dados.

O sensor de temperatura permite obter a temperatura atual do sistema, permitindo detetar o aumento anormal desta medida. O componente escolhido para este sistema foi o MCP9700

³SubMiniature version A

⁴Inter-Integrated Circuits

do fabricante Microchip.

O detetor de vandalismo é um sensor de infravermelhos composto por um LED e um fotorrecetor de infravermelhos que deteta a abertura indevida do invólucro da eletrónica do sistema, gerando um alerta. O componente escolhido para desempenhar esta função no sistema foi o HSDL-9100 do fabricante Broadcom.

A placa-mãe dispõe de 4 conetores laterais para entrada de alimentação e sinais dos sensores e saída de sinais de controlo para os atuadores, para além dos dois conetores de 28 pinos utilizados para a interação com a placa de comunicações.

3.2.2 Placa de comunicações

A placa de comunicações, ilustrada pela figura 3.4, é o elemento de *hardware* do sistema que aloja os componentes de comunicação com o servidor, de leitura dos cartões RFID do utilizador e de interação com o ecrã LCD.



Figura 3.4: Placa de comunicações

A comunicação com o servidor é realizada através de um modem Huawei MU509, que se encontra a comunicar com o sistema através de uma interface UART. Esta ligação consiste num *socket* TCP⁵ entre o modem e o servidor, sendo toda a troca de dados entre estes dois elementos exclusivamente efetuada através dessa interface.

A leitura de cartões do utilizador é efetuada através de um leitor SL031 da StrongLink, que comunica com o microcontrolador através de uma interface UART. A aproximação de um cartão do leitor despoleta no microcontrolador uma interrupção, através de uma saída dedicada do módulo, que origina a leitura do cartão RFID. Este módulo de interface com o cartão RFID permite a leitura e escrita dos mesmos, permitindo a diferenciação entre um

⁵Transport Control Protocol

cartão de utilizador e um cartão de um membro da equipa técnica, ao mesmo tempo que permite atualizar os dados escritos nas memórias desses cartões.

A interação com o ecrã LCD é efetuada através de uma interface SPI cuja taxa de transferência é de 50 MHz. Este valor de taxa de transferência é necessário, de forma a que a atualização de todos os pixels do ecrã seja efetuada a uma frequência de refrescamento tal que não provoque arrasto na imagem, evitando anomalias na imagem que possam ser perturbadoras para o utilizador.

A placa de comunicações interage com a placa-mãe através de dois conetores de 28 pinos existentes em ambas as placas. Este tipo de conetores permite a montagem em sanduíche, permitindo uma montagem mais compacta, que ocupa menos área, quando comparada com uma solução contendo apenas uma placa de circuito impresso.

3.3 PLATAFORMAS DE GESTÃO

Como parte da interface de gestão fornecida ao gestor da implementação do sistema no terreno, a plataforma de gestão do sistema Bikeemotion consiste numa implementação de um *backoffice web* que permite efetuar todo o controlo e monitorização do sistema.

A partir desta plataforma, é possível localizar todos os veículos em tempo real, verificar as rotas atuais e as realizadas previamente, atualizar o *firmware* das bicicletas à distância, listar as docas existentes, assim como permite verificar o estado da sua ocupação, listar os quiosques existentes e verificar se existem falhas, realizar a gestão dos utilizadores, quer sejam pessoas da equipa técnica de manutenção ou os utilizadores que são membros da implementação à qual pertence a plataforma.

3.4 BICICLETA

A bicicleta Bewegen, ilustrada na figura 3.5, corresponde à unidade circulante do sistema Bikeemotion. Este é o elemento cedido ao utilizador como parte da utilização do sistema.

Este elemento, derivado da sua utilização por um grupo vasto de pessoas, é construída numa liga de alumínio bastante robusta, capaz de suportar cargas elevadas sem se danificar. Por outro lado, o seu motor elétrico permite proporcionar ao seu utilizador uma experiência em estradas com bastante inclinação semelhante à que este teria numa estrada plana.

Quando comparada com uma bicicleta tradicional, a bicicleta do sistema Bikeemotion apresenta-se mais pesada, robusta e resistente, ao mesmo tempo em que consegue incorporar toda a eletrónica e baterias no corpo da bicicleta, sem provocar ao utilizador nenhum desconforto na utilização, uma vez que toda a passagem de cabos elétricos é realizada na coluna central da bicicleta e o alojamento da eletrónica na zona central do guiador da mesma.



Figura 3.5: Bicicleta Bewegen[15]

3.5 MODELO DE FUNCIONAMENTO

O Sistema Bikeemotion é uma solução tipicamente utilizada num modelo de funcionamento por assinatura, em que é atribuído ao utilizador um cartão de membro, que este utiliza para desbloquear uma bicicleta quando assim o desejar. Ao realizar este desbloqueio, a viagem realizada é cobrada consoante o plano de membro contratualizado, que contém um valor de distância ou tempo para este utilizar, sendo os pagamentos realizados mensal ou anualmente, acrescido dos períodos realizados por este fora do seu plano de assinatura.

Apesar deste modelo de assinatura, o sistema permite que sejam criadas viagens individuais num quiosque de levantamento, sendo atribuída uma bicicleta, desbloqueada pelo servidor. Em todos os casos de desbloqueio, a conta associada ao utilizador que a desbloqueou contém dos dados do cartão de crédito do mesmo, sendo usada para cobrar os danos que forem realizados no sistema durante a viagem.

3.6 AVALIAÇÃO DO SISTEMA

Após a análise de toda a implementação do sistema, fez-se uma avaliação do estado atual do projeto e efetuou-se um exercício de identificação dos pontos que se apresentam como limitações do sistema. Após este trabalho, ponderou-se qual a melhor solução para cada um desses pontos.

Em primeiro lugar, destacou-se como limitação a existência de apenas uma interface de comunicação com o exterior atualmente em uso, o que impede que se consiga obter o estado de um veículo caso não haja uma ligação de dados ativa.

Em segundo lugar, observou-se que a maioria das operações de manutenção a bicicletas com avarias obriga a uma desmontagem parcial pelo simples fato de não ser possível efetuar o diagnóstico sem recorrer a esse procedimento.

Para resolver a primeira situação, a opção mais viável seria a implementação de uma nova interface de comunicação sem fios, que fosse de curto alcance, permitindo avaliar o estado dos veículos.

Para resolver a segunda situação, aproveita-se a solução apresentada no ponto anterior e desenvolve-se um mecanismo de diagnóstico que possa ser facilmente transportado pela equipa técnica de manutenção.

Trabalho desenvolvido

Após a identificação das lacunas atuais existentes no sistema Bikeemotion, deu-se início ao planeamento e execução das alterações que se pretendem efetuar. O planeamento do projeto consistiu na análise de requisitos do mesmo e identificação clara dos mecanismos necessários para suportar as funcionalidades pretendidas. O desenvolvimento foi dividido em duas partes, sendo que, numa primeira fase, desenvolveu-se uma prova de conceito, cujo propósito foi o de testar se a comunicação sobre Bluetooth Low Energy cumpria os requisitos necessários e se seria possível estabelecer uma ligação bidirecional, com troca de comandos. Esta prova de conceitos consistiu na implementação de um módulo de comunicações BLE numa placa-mãe previamente existente e no desenvolvimento de uma aplicação Android simples, capaz de descobrir, ligar e trocar dados com o módulo de comunicações. Numa fase posterior, procedeu-se à alteração do esquemático e do layout da placa atualmente existente, inserindo-se o módulo de comunicações RN4020, da Microchip, assim como um mecanismo de gestão de alimentação para o mesmo, por forma a controlar o consumo energético.

4.1 REQUISITOS DO DESENVOLVIMENTO

Como analisado no capítulo 3, existem algumas limitações na eficiência que é possível ter durante a manutenção do sistema, principalmente no diagnóstico de problemas que exigem a desmontagem parcial das bicicletas. Por outro lado, apenas existe uma interface de comunicações na bicicleta, que funciona exclusivamente para comunicações com o servidor, que em caso de falha dificulta o diagnóstico das bicicletas, que atualmente só pode ser feito através da plataforma de gestão.

Assim sendo, esta deteção de problemas nas bicicletas poderia ser agilizada caso existisse um mecanismo interno de diagnóstico através de um segundo canal de comunicação, como o Bluetooth Low Energy, e que permitisse o envio de ações e o teste de todos os elementos da bicicleta.

Como resposta a estas necessidades, este trabalho tem como requisitos:

- **Implementação de um módulo de comunicações Bluetooth Low Energy:**
este requisito permite que seja acrescentado um canal de comunicações, que pode ser utilizado por técnicos ou utilizadores finais, funcionando como um mecanismo que pode ser utilizado futuramente para novas funcionalidades;
- **Desenvolvimento de uma camada de aplicação para gestão de mensagens:**
uma vez que as mensagens sobre uma ligação Bluetooth Low Energy não podem ser enviadas em pacotes superiores a 20 bytes, é necessário fazer a divisão e posterior aglomeração, por forma a ser possível interpretar os dados que são enviados e recebidos;
- **Desenvolvimento de um protocolo de comandos:**
uma vez que a comunicação sobre a ligação Bluetooth Low Energy é enviada sobre a forma de bytes de dados, é necessário estabelecer um protocolo que seja interpretado por ambos os terminais, de forma a ser possível desenvolver funcionalidades capazes de tomar partido deste canal de comunicação;
- **Desenvolvimento de uma aplicação Android:**
após toda a integração do módulo RN4020 no *hardware* da placa-mãe, é necessário implementar na aplicação Android existente para equipas técnicas as camadas aplicacionais necessárias para efetuar a ponte entre os dois sistemas.

4.2 PROVA DE CONCEITO

Por forma a avaliar a viabilidade da implementação da tecnologia de comunicação sem fios Bluetooth Low Energy, iniciou-se o trabalho com uma prova de conceito, que consistiu numa implementação de um módulo de comunicações BLE numa placa de circuito impresso já existente. Para fazer o teste ao funcionamento da comunicação BLE, foi desenvolvida uma aplicação Android com as funcionalidades mínimas para se ligar ao módulo e enviar pequenas mensagens.

4.2.1 Hardware

O módulo de comunicações BLE escolhido foi o RN4020 da Microchip, derivado do seu suporte nativo ao protocolo MLDP¹, que simula uma comunicação UART sobre um canal de comunicações BLE. Por outro lado, este módulo já se encontra equipado com as duas camadas de *hardware* necessárias ao BLE, a camada de anfitrião (Host) e de controlador (Controller), o que permite que toda a *stack* BLE esteja já integrada no módulo, sendo todo o controlo e configuração da comunicação efetuada através de comandos enviados pela interface UART.

A placa de circuito impresso fornecida para esta implementação foi uma placa-mãe da Bikeemotion versão 3.9, cedida pela Micro I/O. Esta placa possui um microcontrolador PIC32MZ da Microchip, cuja quantidade de memória disponível excede largamente a necessária para este passo, ao conter 1MB de memória para código executável e 512KB de memória RAM. Da mesma forma, esta placa tem um módulo de comunicações LoRa que não foi montado, pelo que a sua localização foi a ideal para a integração do módulo BLE.

¹Microchip Low-energy Data Protocol

Ao analisar o esquemático da placa-mãe, observou-se que o periférico de UART número 5 do microcontrolador não se encontrava a ser necessário, o que permitiu que existisse uma interface para comunicação com o módulo sem ser necessário efetuar alterações à placa.

Por forma a facilitar a integração do módulo BLE na placa-mãe, escolheu-se o máximo de sinais de controlo de forma a coincidir com os usados pelo módulo LoRa, reduzindo o número de adaptações necessárias nas pistas da placa de circuito impresso, resultando nas ligações apresentadas na figura 4.1.

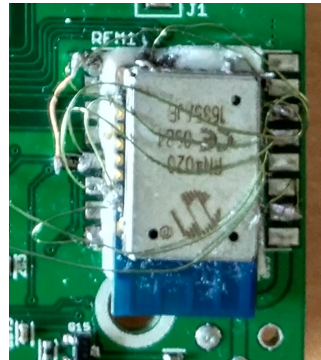


Figura 4.1: Implementação do módulo na prova de conceitos

4.2.2 Firmware

Com a adição do novo módulo de Bluetooth RN4020 da Microchip, foi necessário desenvolver uma nova componente de *firmware*, por forma a ser possível comunicar com o módulo e controlar a execução das transferências de dados entre este e o microcontrolador.

A componente de *firmware* desenvolvida para o controlo básico do módulo Bluetooth consiste em três blocos elementares que desempenham funções claras e objetivas: controlo de alimentação e estado de energia do módulo, gestão da comunicação através da UART com controlo de fluxo por hardware, e uma camada de aplicação, responsável pela receção, envio e processamento de informação recebida pelo microcontrolador. É este bloco que processa, nesta fase inicial, os comandos que chegam da aplicação Android, sendo ele o responsável por executar as instruções que lhe são comunicadas.

A unidade de controlo de energia é responsável por ativar o módulo, gerir a interface de controlo de estado do mesmo, através do controlo das entradas de desativação de hardware e software presentes no dispositivo, assim como a identificação dos períodos em que é necessário desligar o transmissor Bluetooth ou proceder ao seu reinício, caso haja a necessidade de o colocar num estado predefinido.

O bloco de comunicação entre o microcontrolador e o módulo RN4020 consiste numa UART com controlo de fluxo por hardware, um requisito necessário para se poder utilizar o protocolo MLDP desenvolvido pela Microchip. Este é o protocolo que permite a simulação do funcionamento de uma porta série através de uma ligação BLE, através do qual é possível estabelecer uma ligação transparente entre os dois dispositivos, sendo que todos os comandos e notificações entre os dispositivos é efetuado recorrendo a uma codificação ASCII².

²American Standard Code for Information Interchange

Para finalizar, a aplicação de controlo é responsável pela receção e envio de informação, assim como o processamento dos dados recebidos e a interpretação de comandos, sempre que estes são recebidos. Toda a informação remetida pelo módulo que serve para notificação de eventos é filtrada, sendo apenas transmitidos os dados para as outras camadas do *firmware*.

4.2.3 Aplicação Android

Para se proceder aos testes de funcionamento da troca de comandos através da camada de Bluetooth Low Energy, foi necessário desenvolver uma aplicação de teste, capaz de executar todas as funcionalidades especificadas inicialmente. Esta aplicação, de design muito simplista, serviu para confirmar o correto funcionamento da prova de conceito, assim como serviu de base ao desenvolvimento da aplicação final, que acompanha este trabalho.

Esta aplicação é composta por duas atividades, ilustradas pelas figuras 4.2 e 4.3, que implementam os mecanismos detalhados no capítulo 5. Esta aplicação permite fazer a pesquisa de dispositivos nas imediações, estabelecer ligação e enviar comandos de bloqueio e desbloqueio da bicicleta e pedidos de informação. Os mecanismos implementados nesta aplicação de teste foram migrados para a aplicação final durante o desenvolvimento.

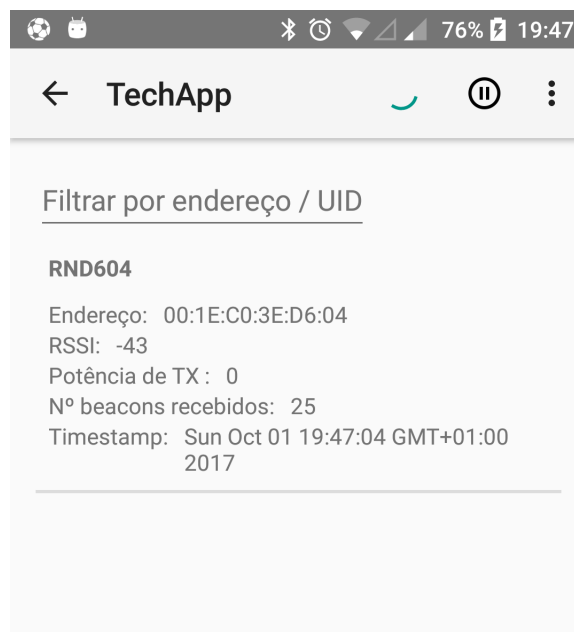


Figura 4.2: Ecrã de pesquisa de dispositivos da aplicação de teste

Esta atividade apresenta algumas informações físicas do dispositivo que foi encontrado, nomeadamente o seu endereço, o RSSI³ comunicado e o número de *beacons* recebidos.

³Received Signal Strength Indication

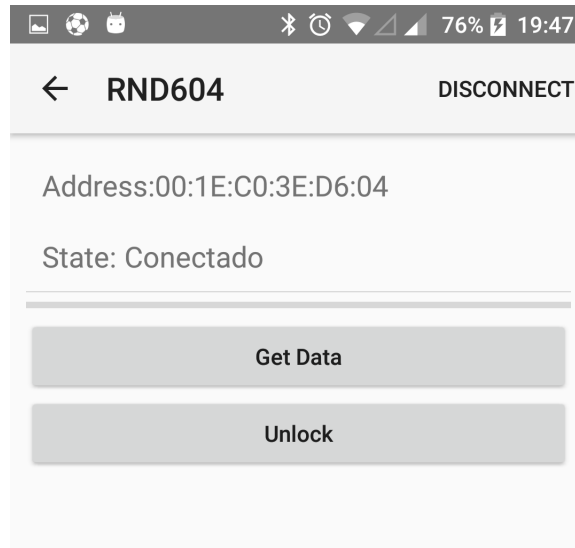


Figura 4.3: Ecrã de comandos da aplicação de teste

4.3 HARDWARE

Para este trabalho, foi escolhida a tecnologia Bluetooth, mais propriamente a sua versão de baixo consumo energético, por duas grandes razões: a primeira razão é o seu baixo consumo energético quando comparado com outras tecnologias de comunicação a curto alcance, o que se torna vital para um sistema como o da Bikeemotion, que funciona sobre fontes de energia não permanentes, ou seja, de baterias; a segunda razão prende-se com a existência de suporte para a tecnologia Bluetooth, e BLE, em quase todos os dispositivos móveis existentes atualmente, o que facilita a sua integração e serve de mais valia para aumentar a utilização da plataforma.

Como referido anteriormente, o módulo de *hardware* escolhido para a comunicação por BLE foi o RN4020 de Microchip. Este módulo foi escolhido por ser uma implementação completa de todas as camadas de hardware necessárias, a de anfitrião e a do controlador, ao mesmo tempo que implementa a *stack* completa de Bluetooth no *firmware* do seu microcontrolador interno. Ao ter estas rotinas já implementadas internamente, este módulo permite integrar o suporte de comunicações BLE no sistema já existente, ao mesmo tempo que permite uma abstração dos mecanismos de comunicação propriamente ditos, uma vez que o *hardware* realiza a ponte entre o dispositivo e a UART que é utilizada para comunicar. Isto significa que, do ponto de vista do *firmware* do sistema das bicicletas, a comunicação BLE é apenas uma ligação UART.

O esquemático do circuito utilizado para a implementação deste módulo na placa previamente existente baseia-se no circuito recomendado pelo fabricante, ilustrado na figura 4.4 e retirado do *Data Sheet* do módulo[37].

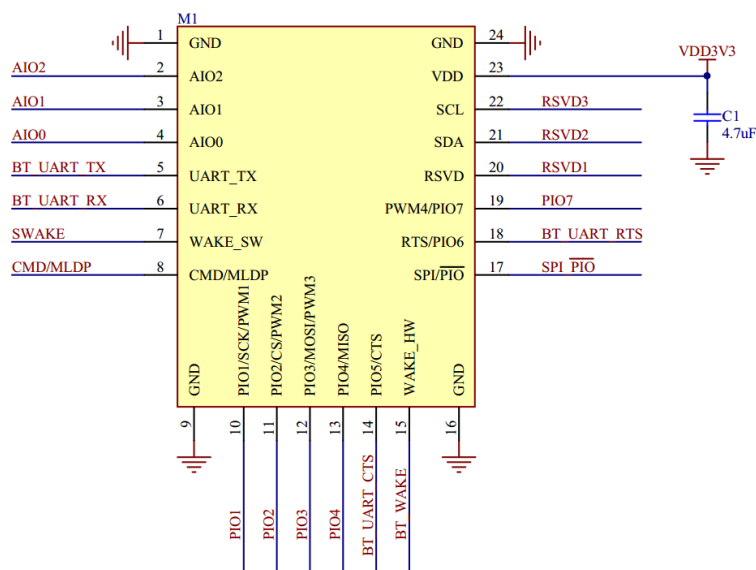


Figura 4.4: Referência para desenho de esquemático do módulo RN4020

Este esquemático foi adaptado às condições ótimas para o novo sistema, através de sucessivos testes de manipulação dos valores dos componentes, por forma a reduzir o ruído nas linhas de alimentação, dando origem ao esquemático apresentado na figura 4.5

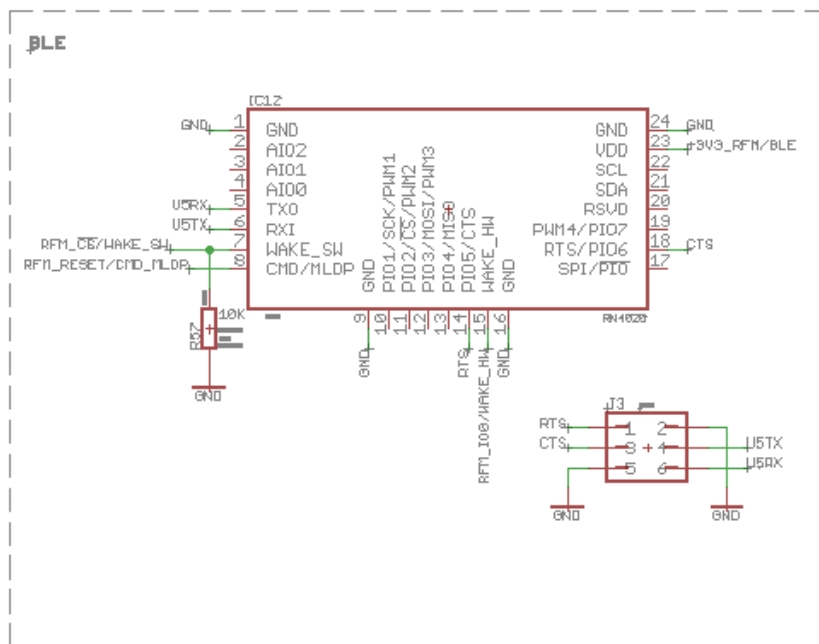


Figura 4.5: Esquemático da implementação BLE

Para o desenvolvimento do layout da placa final, seguiu-se o guia de desenho de *hardware* do fabricante, cumprindo-se com as normas de desenho especificadas por este, como ilustrado na figura 4.6.

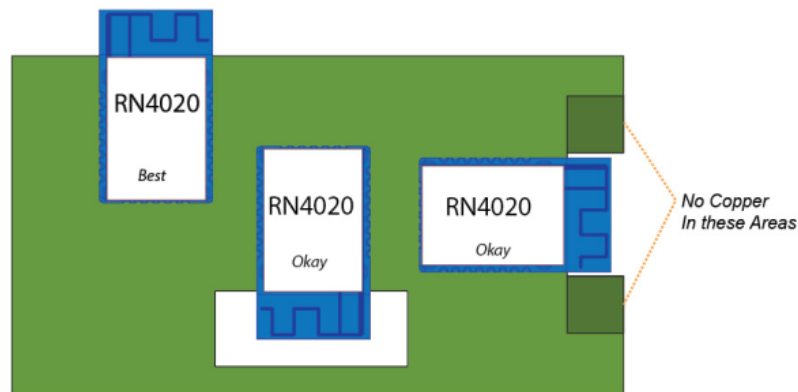


Figura 4.6: Referência para desenho de layout do módulo RN4020

O resultado deste desenho corresponde ao representado pela figura 4.7, que demonstra a secção da placa mãe onde o módulo foi implantado.

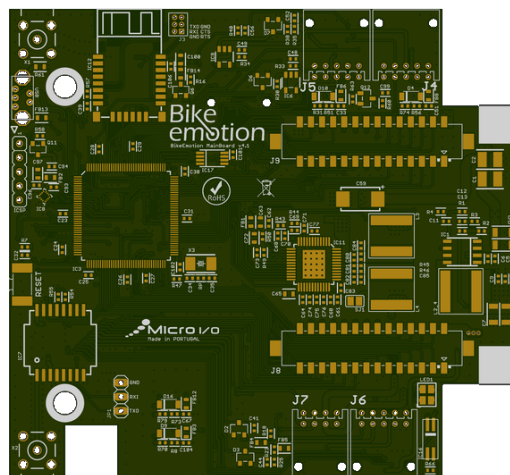


Figura 4.7: Placa de Circuito Impresso com a implementação do módulo RN4020

A opção escolhida para o desenho do *layout* foi a de embutir todo o módulo na área da placa mãe, realizando-se a remoção da área de cobre que se encontraria por baixo da antena do mesmo. Por motivos de limitação de espaço, as áreas laterais da antena tiveram que ser mantidas com cobre, reduzindo-se a sua remoção ao mínimo aceitável. Esta é a solução que aparenta ser mais robusta para o ambiente em que estas placas irão ser inseridas, impedindo quebras originadas por tensões aplicadas dentro do encapsulamento.

As ligações entre as entradas e saídas do módulo RN4020 e do microcontrolador seguem o representado na tabela 4.1, sendo representadas todas as ligações, inclusive as que não foram utilizadas.

ID	Interface do RN4020	Interface do microcontrolador
1	GND	GND
2	AIO2	Não ligado
3	AIO1	Não ligado
4	AIO0	Não ligado
5	UART TX	RC14
6	UART RX	RC13
7	WAKE SW	RE1
8	CMD/MLDP	RE0
9	GND	GND
10-13	PIO1-4	Não ligado
14	UART CTS	RK4
15	WAKE HW	RJ7
16	GND	GND
17	SPI/PIO	Não ligado
18	UART RTS	Não ligado
19	PIO7	Não ligado
20-22	RSVD1-3	Não ligado
23	VDD	VDD_BLE
24	GND	GND

Tabela 4.1: Ligações entre portos do módulo RN4020 e do microcontrolador

O protocolo de comunicação escolhido para as transmissões sobre BLE corresponde ao MLDP, uma especificação proprietária da Microchip, suportada pelo módulo de comunicações RN4020, que permite a comunicação sobre uma ligação Bluetooth de forma simplificada, através de comandos ASCII enviados através da ligação UART. Este protocolo, segundo o guia para o utilizador fornecido[38], especifica a obrigatoriedade da implementação de uma UART com controlo de fluxo por *hardware*. Este requisito não pode ser cumprido na implementação deste projeto, por inexistência de portos de entrada e saída disponíveis no microcontrolador, pelo que será necessário efetuar uma análise do impacto desta alteração na implementação.

Uma vez que foi assumida como taxa de transferência para UART o valor de 115.2 kbits por segundo, e sabendo que a taxa de transferência máxima suportada pelo módulo RN4020 é de 2Mbps, entende-se que, ao ser assegurado um *buffer* de dados suficientemente grande na receção de dados no microcontrolador, apenas seria estritamente necessário o controlo de fluxo no sentido de obter os dados de CTS⁴ do módulo, minimizando-se o número de interfaces que é necessário disponibilizar, resultando nas ligações apresentadas na tabela 4.1.

⁴Clear To Send

Com a organização obtida até ao momento, a comunicação com o módulo BLE encontra-se garantida, permitindo ao microcontrolador saber quando pode enviar dados para o RN4020. A entrada de RTS⁵ deste módulo é deixada em alta impedância, ou seja, não é ligada a nenhuma linha de comunicações. Ao ser mantida uma taxa de transferência bastante inferior aos 2 Mbps máximos do módulo, reduz-se a ocorrência de tentativa de escrita excessiva de dados no mesmo, não ocorrendo perdas de informação.

Após todo o trabalho de desenvolvimento de *hardware*, obteve-se uma placa de circuito impresso com uma atualização dos elementos, tendo agora o módulo BLE previamente referido, um novo recetor GPS e um novo microcontrolador. Estes elementos foram alterados devido a alterações nas cadeias de fornecimento e à necessidade de corrigir certos problemas que foram sendo detetados.

O microcontrolador utilizado inicialmente, da gama ECM[39] da família MZ dos microcontroladores de 32bits do fabricante Microchip, foi lançado com um conjunto elevado de erros de desenho que impediam a utilização em pleno de grande parte dos periféricos, sendo que para alguns dos problemas foram fornecidas sugestões para contornar as ocorrências, mas noutros foi sugerida a completa inutilização desses periféricos, como é observável pelo documento de errata existente[40]. Derivado deste problema, optou-se pela sua substituição por um microcontrolador da gama EFM[41], da mesma família, que corrige alguns desses problemas e traz uma unidade de cálculos em vírgula flutuante que pode ser útil na agilização dos cálculos.

Por outro lado, o módulo de receção GPS utilizado foi colocado em fim de vida pelo fabricante GlobalTop, sendo substituído pelo novo módulo Lady Bird, que suporta, para além do GPS, também as constelações de GLONASS e Galileo, o que permitirá uma localização mais precisa e baseada num maior número de satélites em simultâneo.

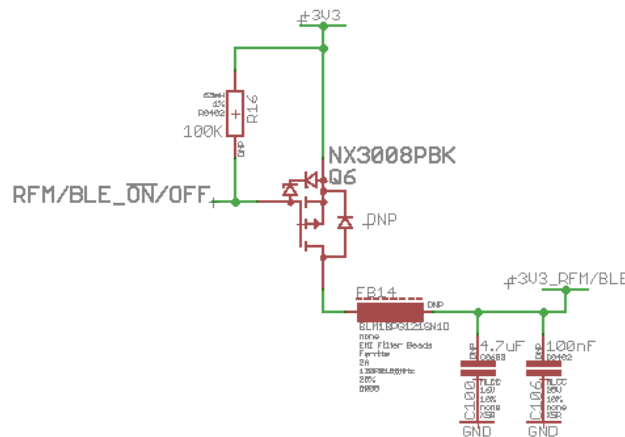


Figura 4.8: Esquemático do circuito do controlo de alimentação do módulo BLE

Para lidar com o controlo de alimentação do módulo de comunicações Bluetooth, foi desenhado um circuito de interrupção da alimentação, cujo esquemático se encontra repre-

⁵Ready To Send

sentado na figura 4.8, que consiste num transístor MOSFET⁶ ligado numa montagem de emissor comum, responsável pelo fornecimento de potência ao módulo. A base deste transístor é controlada pelo microcontrolador, o que permite que este seja desligado no caso de ser necessário reduzir os consumos de energia da bateria.

4.4 FIRMWARE

O *device driver* de BLE consiste num módulo de comunicações UART. Este módulo de *firmware* encontra-se dividido em três partes, sendo elas a inicialização, a operação e a resposta a interrupções.

A inicialização do módulo consiste na configuração dos registos do periférico, nomeadamente a taxa de transferência, a prioridade da interrupção, a ativação das interrupções, a inicialização dos *buffers* circulares de leitura e escrita, e a ativação dos módulos de receção e transmissão do periférico.

A operação do *device driver* consiste nos métodos de escrita e leitura dos dados em modo *polling* e assíncrono, por intermédio da rotina de tratamento da interrupção, e na rotina de tratamento de tarefas onde é executado todo o processamento de dados que foram recebidos por interrupção e os que estão agendados para ser enviados.

A resposta a interrupções corresponde a uma rotina de resposta à interrupção que gere a leitura e escrita de dados na UART, por interrupção. Este mecanismo permite o envio e receção de dados de forma assíncrona, não sendo necessário despendar de processamento adicional caso não haja dados a entrar neste periférico.

Por forma a efetuar a gestão de energia deste módulo de forma eficiente, foi necessário implementar uma sequência de código para o controlo de energia, nomeadamente a atuação da saída do microcontrolador que ativa o transístor de alimentação do módulo BLE.

Segundo as especificações da norma Bluetooth Low Energy, o tamanho máximo dos dados que podem ser enviados em cada mensagem está limitado a 20 bytes. Assim sendo, é necessário efetuar a divisão de mensagens que superem esse valor em pacotes intermédios de valor igual ou inferior a este limite. Desta forma, para conseguir este objetivo, foi necessário desenvolver um gestor de mensagens que fosse capaz de efetuar essa mesma quebra nas mensagens, ao mesmo tempo que indexa os segmentos, permitindo o seu acesso uma ou várias vezes, através de pedidos enviados ao processador de comandos da API, como detalhado no capítulo 6. Este mecanismo foi desenhado por forma a permitir a troca de mensagens e execução de comandos previamente existentes na plataforma das bicicletas, assim como novos comandos desenvolvidos para proporcionar novas funcionalidades, segundo os requisitos exigidos.

Seguindo a mesma linha de pensamento, para efetuar a gestão dos estados do módulo BLE, foi necessário desenvolver uma camada que implementasse o diagrama de estados representado pela figura 4.9, em que o estado de *initiating* corresponde ao estado de configuração do módulo, o estado de *standby* corresponde ao modo em que o dispositivo não se encontra a aceitar ligações, o estado de *advertising* corresponde ao envio de *beacons* que permitem

⁶Metal Oxide Semiconductor Field Effect Transistor

estabelecer uma ligação e o estado *connection* corresponde ao estado de troca de dados entre o microcontrolador e o dispositivo *Bluetooth Low Energy*.

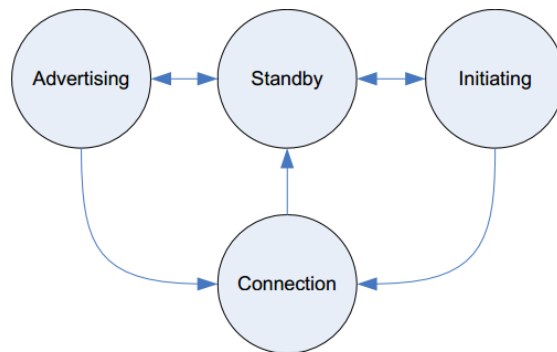


Figura 4.9: Máquina de Estados da aplicação

4.5 GESTÃO DE ALIMENTAÇÃO

Uma vez que este sistema é alimentado por uma fonte de energia finita, proveniente do seu sistema de baterias, e se pretender que a autonomia da bicicleta seja a máxima possível, é necessário ter em conta todo um conjunto de consumos estáticos dos módulos que compõem todo o sistema.

Assim sendo, o consumo dos módulos é gerido consoante o estado em que a bicicleta se encontra. Quando a bicicleta se encontra numa doca de carga, todo o consumo do hardware é minimizado para agilizar o restauro de energia nas baterias. Da mesma forma, quando uma bicicleta se encontra bloqueada, o seu ecrã é desligado e a velocidade do relógio do microcontrolador é reduzida de 200 MHz para 25 MHz, ao mesmo tempo que o módulo RN4020 é mantido em modo de poupança de energia até que a bicicleta seja acordada e se inicie uma ligação Bluetooth Low Energy.

Aplicação Móvel Android

Uma das ferramentas que este trabalho pretende potenciar é a aplicação móvel Android para a equipa técnica do sistema Bikeemotion. Esta ferramenta consiste numa solução completa que permite ao utilizador localizar estações, bicicletas e quiosques, assim como listar os eventos e rotas dos utilizadores registados na base de dados do sistema a que o técnico pertence.

Com a implementação e desenvolvimento das funcionalidades sobre uma ligação Bluetooth Low Energy, foi necessário incluir o suporte a este tipo de ligações na aplicação móvel.

Este capítulo começa por apresentar uma descrição do Android e do funcionamento do seu SDK, apresentando as componentes que fazem parte do desenvolvimento de aplicações, assim como todo o fluxo de execução das diferentes partes da aplicação móvel.

Numa primeira fase, este capítulo descreve todo o funcionamento da aplicação móvel, apresentando os ecrãs existentes, assim como quais as interfaces com o utilizador existentes em cada um deles. Numa segunda fase, é descrito todo o trabalho de integração das tecnologias de comunicação sem fios Bluetooth Low Energy e de controlo das funcionalidades BLE implementadas nas bicicletas, dividido pelos vários elementos que foram adicionados.

O detalhe dado ao desenvolvimento da implementação das funcionalidades que são suportadas pela ligação Bluetooth Low Energy foi bastante mais acentuado do que o detalhe dado ao resto da aplicação, uma vez que foge ao âmbito deste trabalho e não foi possível obter muitas informações em relação ao processo de desenvolvimento realizado previamente para a integração da aplicação existente com os servidores existentes no sistema.

5.1 SISTEMA OPERATIVO ANDROID

O Android[42] é um sistema operativo móvel baseado em Linux e desenvolvido pela Google. Desde o seu aparecimento, em 2008, o Android tem captado o interesse de diversas companhias, criadores de aplicações e dos utilizadores em geral. Esta plataforma de software tem sido constantemente melhorada, quer em termos de novas características, quer em hardware suportado. O Android tem a vantagem de ser uma comunidade open-source e de estar cada

vez mais disponível para um largo espectro de dispositivos. A sua arquitetura consiste em 5 camadas: Aplicações (Applications), Framework de aplicações (Application Framework), Bibliotecas (Libraries), Android Runtime e Linux Kernel, como se observa na figura 5.1. A camada de Aplicações situa-se no topo e oferece um conjunto de aplicações, visíveis ao utilizador, que fazem parte do conjunto essencial de qualquer sistema operativo móvel, como mensagem de texto e multimédia (SMS/MMS), relógio, calculadora, correio eletrónico, entre outros. A camada seguinte, Framework de Aplicações, fornece as APIs necessárias para serem usadas pelas aplicações da camada anteriormente descrita. Abaixo desta camada existe uma outra, Bibliotecas, que fornece as bibliotecas essenciais de suporte para as aplicações, como as de C/C++, multimédia, base de dados e desempenho gráfico. Por sua vez, esta camada engloba uma outra, Android Runtime, que fornece a sua própria máquina virtual (VM¹), denominada ART², nas versões de Android iguais ou superiores à 5.0 Lollipop, e DalvikVM, nas versões que antecedem esta, cuja pilha é baseada na Java VM, além de fornecer de igual modo outras bibliotecas de suporte às aplicações (Core Libraries). Por fim, a camada mais inferior diz respeito ao kernel Linux, que contém os controladores de Wi-Fi, Bluetooth, teclado, áudio, memória, câmara, entre outros.

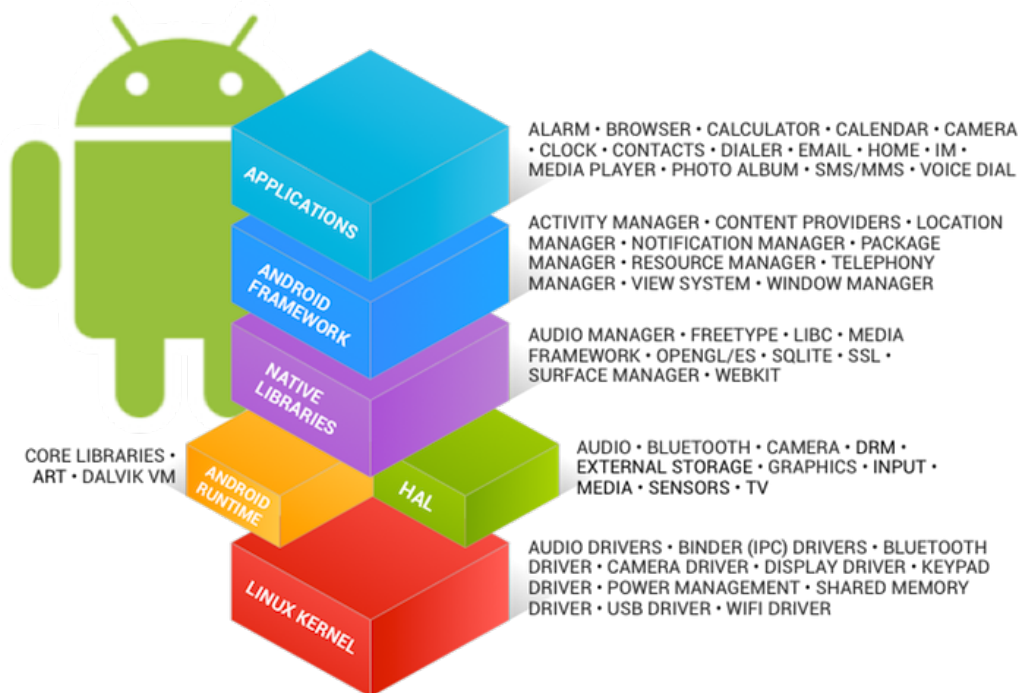


Figura 5.1: Arquitetura do Android[43]

Atualmente, o Android encontra-se na versão 8.1 Oreo. A partir da versão 4.3 Jelly Bean foi possível o suporte para dispositivos Bluetooth Smart Ready, ou seja, incorporação da tecnologia BLE. Toda a documentação e APIs acerca do BLE está disponível na página destinada aos criadores de aplicações Android[44]. O Android, dada a sua versatilidade e

¹Virtual Machine

²Android Runtime

como plataforma open-source, tem captado cada vez mais a atração dos utilizadores comuns, afirmando-se assim no segmento dos dispositivos móveis. Segundo a IDC³, entidade de referência para estudos de mercado na área das tecnologias, no primeiro trimestre de 2017, apresentaram uma quota de mercado na ordem dos 85 %[45].

5.1.1 Detalhes da Plataforma Android

Uma aplicação desenvolvida sob a plataforma Android tem 4 componentes essenciais: atividades, serviços, fornecedores de conteúdos e recetores de transmissões. Cada componente serve um determinado propósito e tem um ciclo de vida distinto que define como e quando o componente é criado e destruído. As atividades representam o ecrã de interface com utilizador para que este possa interagir com a aplicação, ou seja, basicamente é o que frequentemente se chama de janela. Uma aplicação pode ter várias atividades, cada uma com tarefas distintas, sendo por isso independentes umas das outras. Por exemplo, uma aplicação de correio eletrónico pode ter uma atividade para listar o correio recebido, outra para ler e outra para escrever. Todas trabalham de forma unida para fornecer a melhor experiência ao utilizador. No Android, as atividades são geridas como uma pilha, em que quando uma nova atividade é iniciada, é colocada no topo da pilha, tornando-se a atividade em primeiro plano. A atividade anterior permanece sempre em segundo plano enquanto a atual existir. O ciclo de vida pode ser explicado pela figura 5.2.

Uma atividade criada e que esteja em primeiro plano é considerada em execução (running). Caso esta perca protagonismo mas ainda assim seja visível, ou seja, se outra atividade que não ocupe todo o ecrã aparecer em primeiro plano, a primeira fica em pausa. Continua a ser uma atividade presente, mas pode ser destruída (killed) a qualquer momento caso o sistema necessite de libertar memória. Caso a atividade seja completamente obstruída por outra, esta fica inativa, em estado de paragem. Continua com o estado atual e com as suas informações, mas já não é mais visível para o utilizador, sendo assim terminada, ou acaba destruída pelo sistema.

³International Data Corporation

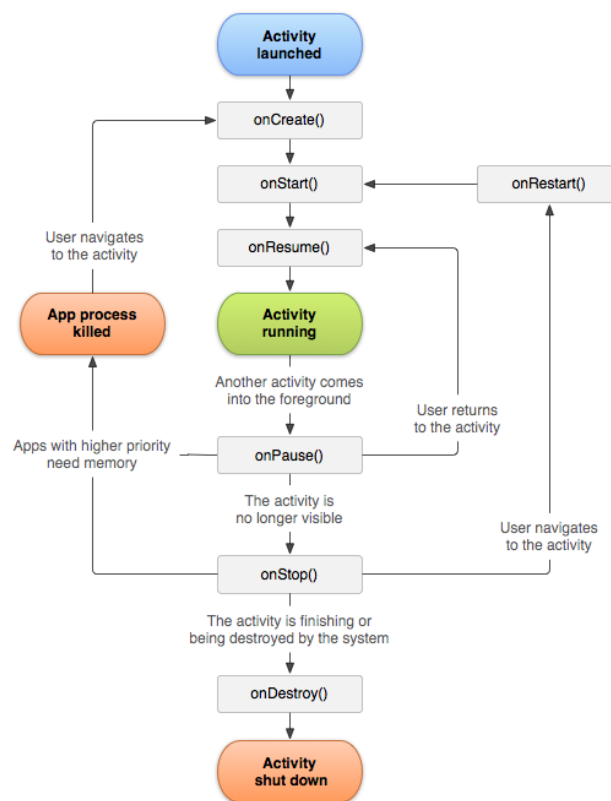


Figura 5.2: Ciclo de vida de uma atividade[46]

Existem métodos que podem ser implementados para desempenharem diversas operações quando a atividade transita entre os diferentes estados, e que, de certa forma, auxiliam o programador a controlar o estado da aplicação. Os métodos são:

- **onCreate()** – é o primeiro método a ser executado no momento em que uma atividade é lançada. É responsável por converter os ficheiros XML de layout em objectos visíveis na aplicação, sendo apenas executado uma vez durante todo o ciclo de vida de uma atividade.
- **onStart()** – é o método invocado após o onCreate(), sendo também invocado quando uma atividade que estava em segundo plano (invisível para o utilizador), volta a ter foco.
- **onResume()** – este método é chamado quando a atividade retorna para primeiro plano, depois de estar parcialmente encoberta por outra atividade. De notar que neste caso a atividade nunca deixou de estar visível para o utilizador, apenas estava em pausa, não em paragem completa.
- **onPause()** – é o método que é executado quando a atividade é parcialmente sobreposta por outra.
- **onStop()** – é o método executado quando uma atividade perde completamente a visibilidade para o utilizador.
- **onDestroy()** – é o ultimo método a ser executado. Depois deste a atividade é considerada destruída, não podendo ser relançada.

- **onRestart()** – é invocado imediatamente antes do **onStart()**, quando a atividade passa novamente para primeiro plano depois de estar invisível ao utilizador.

Os serviços (services) são um tipo de componente que corre em segundo plano, sendo desenhados para realizar operações durante longos períodos de tempo. Um serviço não fornece interface gráfica para diálogo com o utilizador, sendo a sua função executar sem interferir com a ação de outras atividades. Os fornecedores de conteúdos têm a capacidade de gerir um conjunto partilhado de dados das aplicações. É possível guardar os dados de uma aplicação na raiz do sistema Android, na base de dados do sistema (SQLite), na Internet ou em qualquer outro local onde a aplicação tenha permissões de aceder com permissões de escrita. Através dos fornecedores de conteúdos, aplicações podem requisitar ou até modificar dados de outra aplicação, ou seja, são responsáveis por fazer a ponte entre aplicações no que diz respeito à obtenção e troca de informação. Por fim, existem os recetores de transmissões (*broadcast receivers*), que são componentes que permitem a receção de mensagens por parte das outras aplicações ou pelo sistema. Estas mensagens são chamadas de eventos. Iniciar uma chamada, terminar uma chamada, ou receber um SMS, são exemplos de eventos que podem ocorrer e, através do recetor de transmissões, são enviados avisos às aplicações para que estas, se desejarem, possam responder a estes eventos.

5.2 ATIVIDADES

5.2.1 Login

A atividade de login permite fazer o controlo de acessos ao sistema Bikeemotion. Este formulário de autenticação permite a introdução do código do cartão RFID do técnico e do respetivo PIN⁴ pessoal, como se pode observar na figura 5.3.

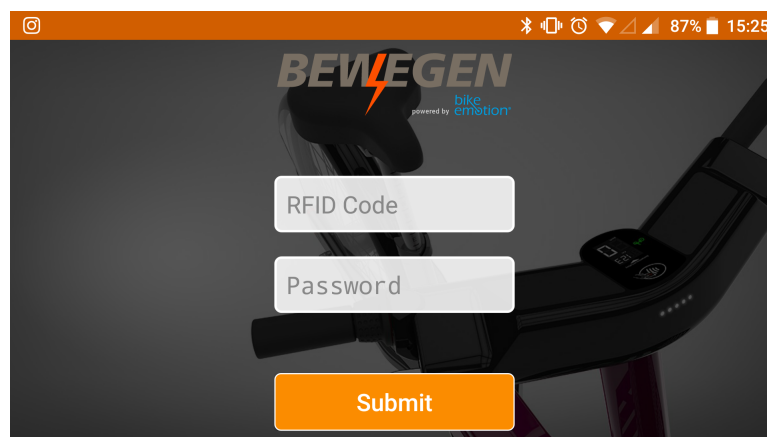


Figura 5.3: Atividade de login

Após uma autorização ser concedida, o sistema passa automaticamente para a atividade do mapa em tempo real.

⁴Personal Identification Number

5.2.2 Mapa em tempo real

O ecrã do mapa em tempo real permite ver uma apresentação gráfica da localização das bicicletas, docas e quiosques, aparecendo o mapa centrado na localização atual do utilizador. Como se pode observar pela figura 5.4, os elementos do sistema aparecem como marcadores no mapa, sendo diferenciados pelo tipo de ícone em cada um deles.

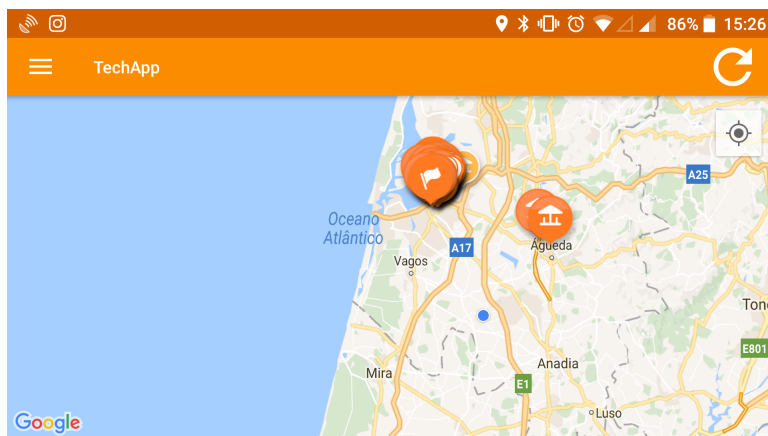


Figura 5.4: Atividade do mapa de tempo real

Ao ser pressionado um desses marcadores, são fornecidas informações adicionais, como o nome do sistema ao qual esse elemento está afeto ou o número de bicicletas existentes, caso seja um ponto de entrega.

5.2.3 Estações

A atividade de listagem de rotas permite obter uma lista de estações existentes no sistema, ao mesmo tempo que permite saber informações como o número de veículos que se encontram alojados, o número de docas existentes e o número de docas que se encontram vazias, como ilustrado na figura 5.5.

A screenshot of a mobile application interface titled 'Stations'. It displays a table with station information. The table has columns for CODE, NAME, NUMBER, DESCRIPTION, STAT US, VEHIC LES, DOCK S, and SPAC ES. There are three rows of data. Below the table, there is a 'SORT BY' section with a dropdown menu currently set to 'Name: Ascendant'. The app's status bar at the top shows connectivity icons, battery level at 85%, and time at 15:26. A hamburger menu icon is on the left, and another hamburger menu icon is on the right.

CODE	NAME	NUMBER	DESCRIPTION	STAT US	VEHIC LES	DOCK S	SPAC ES
STA05496	3WDynamic	3WD	3WDynamics description 8	OPEN	1/0	5	0
STA03011	AveiroExpo	AE	AveiroExpo	CLOSED	1/0	10	5
STA07428	Baltimore Warehouse	BALT	Baltimore Warehouse	OPEN	1/1	2	2

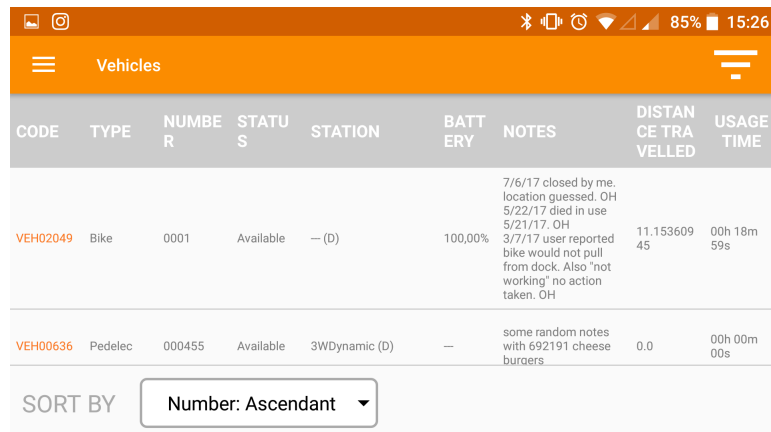
SORT BY Name: Ascendant

Figura 5.5: Atividade de lista de estações

A listagem obtida pode ser ordenada pelo nome, número ou estado da doca, e pode ser filtrada pelo estado, ocupação ou tipo de estação, se é temporária ou fixa.

5.2.4 Veículos

A atividade de listagem de rotas permite obter uma lista de veículos registrados no sistema, ao mesmo tempo que permite saber informações como o código do veículo, a estação a que pertence, a distância já percorrida e o tempo em que esteve em uso, como ilustrado na figura 5.6.



CODE	TYPE	NUMBER	STATUS	STATION	BATTERY	NOTES	DISTANCE TRAVELED	USAGE TIME
VEH02049	Bike	0001	Available	— (D)	100,00%	7/6/17 closed by me, location guessed. OH 5/22/17 died in use 5/21/17. OH 3/7/17 user reported bike would not pull from dock. Also "not working" no action taken. OH	11.15360945	00h 18m 59s
VEH00636	Pedelec	000455	Available	3WDynamic (D)	—	some random notes with 692191 cheese burgers	0.0	00h 00m 00s

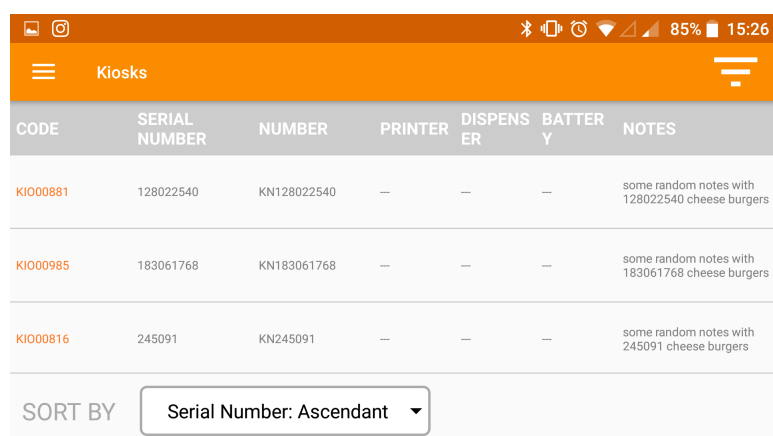
SORT BY Number: Ascendant

Figura 5.6: Atividade de lista de veículos

A listagem obtida pode ser ordenada pelo número ou estado do veículo e pode ser filtrada pela disponibilidade ou indisponibilidade da bicicleta.

5.2.5 Quiosques

A atividade de listagem de rotas permite obter uma lista dos quiosques existentes no sistema, ao mesmo tempo que permite saber informações como o código do quiosque, o número de série, assim como saber a existência de bateria, impressora ou dispensador de cartões, como ilustrado na figura 5.7.



CODE	SERIAL NUMBER	NUMBER	PRINTER	DISPENSER	BATTERY	NOTES
KIO00881	128022540	KN128022540	—	—	—	some random notes with 128022540 cheese burgers
KIO00985	183061768	KN183061768	—	—	—	some random notes with 183061768 cheese burgers
KIO00816	245091	KN245091	—	—	—	some random notes with 245091 cheese burgers

SORT BY Serial Number: Ascendant

Figura 5.7: Atividade de lista de quiosques

A listagem obtida pode ser ordenada pelo número de série.

5.2.6 Eventos

A atividade de listagem de rotas permite obter uma lista dos eventos registrados no sistema, ao mesmo tempo que permite saber informações como a descrição, tipo, estado e data de origem do evento, como ilustrado na figura 5.8.

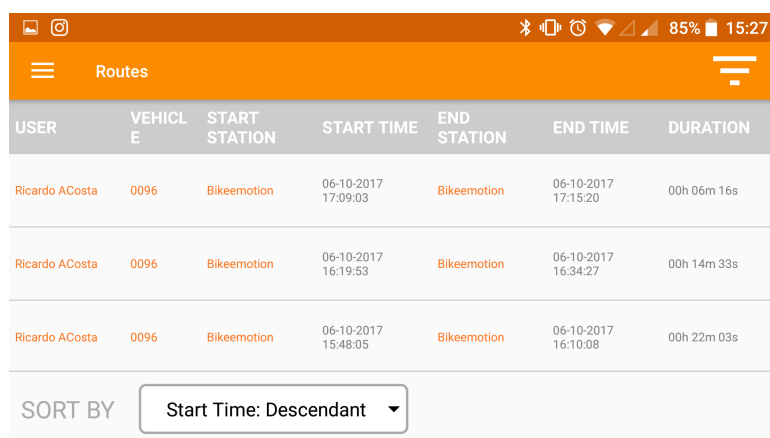
</

Figura 5.8: Atividade de lista de eventos

A listagem obtida pode ser ordenada pela data de criação, pelo tipo, pelo evento e pela descrição e pode ser filtrada pela data de criação do evento, pela data de fecho do mesmo ou pelo conhecimento que já foi tomado pela existência desse evento.

5.2.7 Rotas

A atividade de listagem de rotas permite obter uma lista das rotas registradas no sistema, ao mesmo tempo que permite saber informações como o nome do utilizador que efetuou a rota, o veículo que utilizou, a data e hora da partida e chegada, as estações de levantamento e entrega da bicicleta e a duração da viagem, como ilustrado na figura 5.9.



USER	VEHICLE	START STATION	START TIME	END STATION	END TIME	DURATION
Ricardo ACosta	0096	Bikeemotion	06-10-2017 17:09:03	Bikeemotion	06-10-2017 17:15:20	00h 06m 16s
Ricardo ACosta	0096	Bikeemotion	06-10-2017 16:19:53	Bikeemotion	06-10-2017 16:34:27	00h 14m 33s
Ricardo ACosta	0096	Bikeemotion	06-10-2017 15:48:05	Bikeemotion	06-10-2017 16:10:08	00h 22m 03s

SORT BY: Start Time: Descendant

Figura 5.9: Atividade de lista de rotas

A listagem obtida pode ser ordenada pela data de levantamento e pode ser filtrada pela data e hora do bloqueio e de desbloqueio da bicicleta.

5.2.8 Bluetooth

A atividade que serve de suporte ao Bluetooth Low Energy é dividida em duas partes: uma que efetua a procura dos dispositivos nas proximidades e outra que efetua a ligação e gestão da troca de informações entre o telemóvel e a bicicleta.

A pesquisa de dispositivos é efetuada no ecrã apresentado na figura 5.10, onde aparecem todos os dispositivos que suportem o protocolo de comunicações BLE MLDP. Quando um destes dispositivos é encontrado, é apresentada uma entrada como referida, apresentando-se o seu endereço, a sua potência de sinal, representada pelo parâmetro RSSI, o número de pacotes de *broadcast* recebidos e a data e hora da última receção destes pacotes.



Figura 5.10: Atividade de pesquisa de dispositivos Bluetooth Low Energy

Caso um destes dispositivos seja o pretendido, através de um breve pressionar na entrada da lista correspondente, é possível passar para o novo ecrã, ilustrado na figura 5.11, em que será efetuada a ligação ao dispositivo e poder-se-á enviar comandos de ação para a bicicleta.

Os comandos atualmente disponíveis para ser utilizados na aplicação correspondem a:

- **Bloquear/desbloquear bicicleta:**
este comando permite enviar um pedido de bloqueio e desbloqueio da bicicleta, deixando-a preparada para ser iniciada uma viagem;
- **Pedido de informações:**
este comando permite enviar um pedido de informações à bicicleta, que responde com uma mensagem de diagnóstico correspondente a um ficheiro JSON⁵ contendo o estado de todos os elementos existentes no *hardware* da bicicleta;
- **Pedido de reinício:**
este comando permite pedir ao *hardware* da bicicleta que efetue um reinício, deixando-a num estado conhecido e seguro;
- **Entrada/navegação/saída do menu técnico:**
este comando permite pedir ao *hardware* da bicicleta que entre em modo técnico, permitindo navegar nas diversas páginas de diagnóstico existentes, permitindo este menu mesmo quando o botão de buzina não estiver completamente funcional.

⁵JavaScript Object Notation

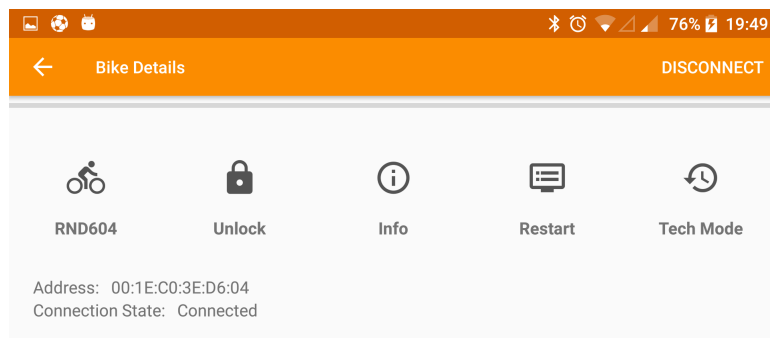


Figura 5.11: Atividade de envio de comandos para as bicicletas

5.3 PERMISSÕES

Nas aplicações Android, as permissões correspondem a autorizações implícitas ou explícitas dadas pelo utilizador quando a aplicação precisa de aceder a um ou mais recursos do sistema que pode conter dados sensíveis, como tecnologias de comunicação, localização por GPS, mensagens recebidas ou contatos telefónicos. Após o lançamento do Android 6, conhecido como M ou Marshmallow, todas as versões lançadas exigiam que as autorizações dadas pelo utilizador fossem explícitas, ou seja, pedidas através de uma janela, ilustrada pela figura 5.12, que é mostrada durante o funcionamento da aplicação, o que permite ao utilizador autorizar o acesso apenas quando este é necessário, podendo revogar as permissões posteriormente, na área de configurações do sistema operativo. Nestes casos, uma vez que a aplicação pode perder as permissões de acesso entre duas execuções, as normas de desenvolvimento de aplicações exigem que sejam verificadas as permissões sempre que um determinado recurso for acedido.

A aplicação desenvolvida neste trabalho necessita de permissões de acesso à Internet, a comunicações Bluetooth e à localização aproximada do utilizador. Esta última permissão é necessária, uma vez que o Bluetooth Low Energy é considerada como uma tecnologia de localização aproximada.

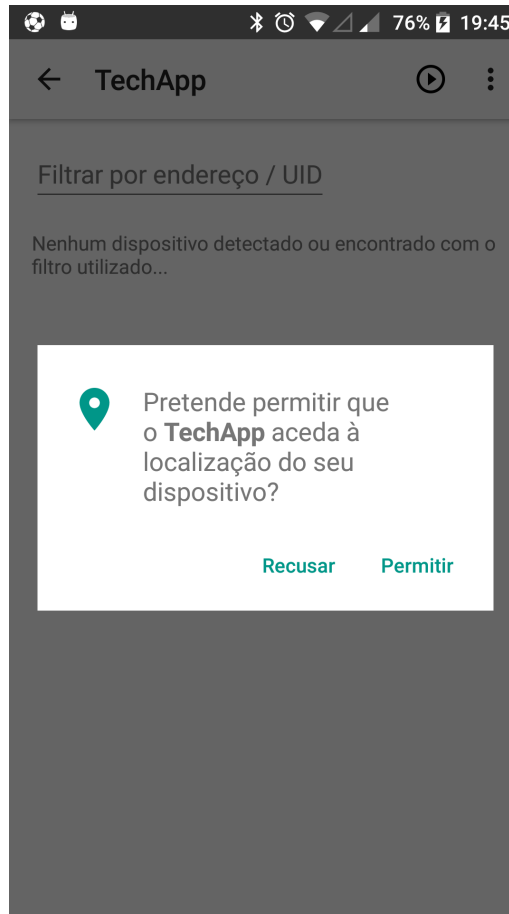


Figura 5.12: Janela de permissões explícitas do sistema Android

5.4 SERVIÇO DE BLUETOOTH LOW ENERGY

O serviço de *Bluetooth Low Energy* é um módulo de *software* desenvolvido para gerir todos os passos da ligação e gestão da interface BLE. Este elemento é o responsável pela identificação dos dispositivos que se encontram ao alcance e verificar se suportam as funcionalidades requeridas pela aplicação, nomeadamente o suporte à versão 4.0 ou superior da norma Bluetooth, que inclui o Low Energy, o suporte ao serviço MLDP e permitirem o estabelecimento de uma ligação a este tipo de dispositivos.

Sempre que o serviço necessita de notificar as outras threads da existência de um evento, este realiza um *broadcast* da mensagem de evento pretendida para todas as outras atividades, que a irão receber se tiverem uma implementação de um *Broadcast Receiver* que as possa processar.

Este serviço suporta a deteção de ligações, da existência de dispositivos ao alcance, de receção da notificação da existência de mensagens que podem ser lidas, do envio bem sucedido das mensagens e da queda da ligação atual, segundo a tabela 5.1, que apresenta a lista de eventos que o serviço despoleta através de um *Broadcast*.

Nome	Descrição
GATT CONNECTED	A ligação foi estabelecida
GATT DISCONNECTED	A ligação foi quebrada
GATT SERVICES DISCOVERED	Foram encontrados serviços suportados pelo dispositivo
DATA AVAILABLE	Há dados para serem lidos
DATA WRITTEN	A mensagem foi enviada com sucesso

Tabela 5.1: Eventos de notificação do serviço BLE

Ao ser iniciado o serviço, é feita uma associação com a atividade que o iniciou, permitindo que o serviço configure os mecanismos de recepção de mensagens, pedindo à camada de *Bluetooth* que o notifique sempre que a escrita seja bem sucedida ou que existam dados disponíveis para ser lidos.

Por forma a ser possível efetuar pedidos ao serviço, este tem um conjunto de métodos que podem ser acedidos pelas restantes camadas da aplicação:

- **connect:** este método permite pedir ao serviço que se ligue ao dispositivo cujo endereço é passado como argumento. Caso esta ligação seja bem sucedida, o serviço efetua um *broadcast* do evento *GATT CONNECTED*;
- **disconnect:** este método permite pedir ao serviço que se desligue do dispositivo em que se encontra atualmente, enviando o evento *GATT DISCONNECTED* caso seja bem sucedida a quebra da ligação;
- **SetCharacteristicNotification:** este método permite pedir à camada *Bluetooth* que notifique da existência de mensagens novas e do correto seguimento das mensagens enviadas;
- **SetCharacteristicIndication:** este método permite pedir à camada *Bluetooth* que notifique da existência de mensagens novas e do correto seguimento das mensagens enviadas, ao mesmo tempo que permite que seja enviada uma mensagem de confirmação da recepção do pacote. Este mecanismo permite efetuar uma ligação sem perda de pacotes, mas provoca um aumento no tempo de transferência, uma vez que a resposta apenas pode ser enviada no bloco temporal seguinte;

Ao ser iniciado um *Scanner BLE*, caso seja encontrado um dispositivo ao alcance, o serviço envia um evento de *SERVICES DISCOVERED* para que seja verificado se o dispositivo suporta a ligação.

5.5 GESTOR DE MENSAGENS

O gestor de mensagens é um módulo de *software* desenvolvido para efetuar a gestão dos pacotes de mensagens enviados e recebidos. Derivado da limitação no tamanho dos pacotes que é possível enviar sobre uma interface Bluetooth Low Energy, que corresponde a 20 bytes por mensagem, foi necessário desenvolver um mecanismo para dividir as mensagens que se

pretende enviar, e para requisitar e agrupar as mensagens parciais que compõem as respostas recebidas, tendo também a capacidade de notificar o número de pacotes que estão disponíveis para serem requisitados para descarga.

Este mecanismo, para facilitar os reenvios, realiza o arquivamento dos pacotes, existindo a implementação de um comando que efetua o pedido de um pacote com um certo número de índice. Assim, caso não haja uma resposta de confirmação a um dos pacotes, este pode ser reenviado.

Aliado a este mecanismo, todos os comandos que sejam enviados e excedam o limite de 20 bytes, são automaticamente divididos em mensagens dentro desse limite, não existindo a indexação referida no ponto anterior.

5.6 PROCESSADOR DE COMANDOS

O processador de comandos é um módulo de *software* desenvolvido para processar todos os comandos que são recebidos pela aplicação através da ligação Bluetooth Low Energy. Este elemento contém uma implementação de todos os comandos descritos no capítulo 6, por forma a ser capaz de reconhecer as respostas recebidas quando cada comando é enviado.

Este módulo foi desenvolvido para permitir uma generalização do funcionamento da aplicação, aglomerando todo o processamento de comandos num módulo único, agilizando a implementação de novas funcionalidades.

5.7 SCANNER BLUETOOTH LOW ENERGY

Por forma a conseguir efetuar a deteção de dispositivo *Bluetooth Low Energy* nas proximidades, é necessário implementar um *Scanner* BLE. Ao trabalhar em conjunto com o serviço, os domínios de atividades dos dois mecanismos é separado, sendo o *scanner* utilizado para fazer a pesquisa de dispositivos nas proximidades e o serviço utilizado para efetuar a ligação e gerir as comunicações após esse momento.

O *Scanner* BLE, derivado da sua natureza integrada nos pacotes do SDK do Android, obriga a que exista a implementação de um conjunto de métodos utilizados para efetuar a notificação de um evento ocorrido durante a pesquisa, nomeadamente:

- OnScanResult: este método é chamado sempre que é detetado um novo dispositivo nas imediações;
- OnScanFailed: este método é chamado sempre que a pesquisa por dispositivos falhar;

Ao ser chamado o método OnScanResult, é fornecido como argumento um elemento que transporta as informações do dispositivo encontrado, sendo utilizados para apresentar ao utilizador, numa lista, como é possível ver na figura 5.10, mostrada anteriormente.

Ao ser chamado o método OnScanFailed, é fornecido o código de erro que originou a falha, ficando registado no histórico de ocorrências da aplicação.

5.8 BROADCAST RECEIVER

O *Broadcast Receiver* é um mecanismo nativo das aplicações Android que suporta a comunicação entre diferentes *threads*, semelhante ao mecanismo de IPC⁶ existente no sistema operativo. Este elemento pode ser implementado em todas as atividades que desejem trocar mensagens ou receber notificações de outras atividades. Este mecanismo também está sujeito às regras das permissões, não sendo permitido aceder a mensagens de um recurso para o qual a aplicação não tem permissões.

Ao receber uma mensagem, o *Broadcast Receiver* pode despoletar ações diferenciadas entre mensagens. A implementação que foi realizada neste trabalho permite detetar a existência de novas mensagens recebidas e o sucesso no envio de mensagens. O primeiro caso despoleta uma leitura da mensagem recebida, enquanto que o segundo caso autoriza o processador de mensagens a enviar o próximo pacote.

⁶Inter-Process Communication

Protocolo de Comunicação

Com o desenvolvimento deste trabalho, tornou-se necessário ter um mecanismo de troca de mensagens que fosse capaz de, ao mesmo tempo, ser flexível e cumprir todos os requisitos especificados. Dessa forma, desenvolveu-se um conjunto de especificações que permitissem à aplicação comunicar com o microcontrolador, efetuando a troca de mensagens tanto de dados como de comandos. Assim sendo, este capítulo descreve a estrutura dos comandos utilizados, assim como todos os já existentes nos dois subsistemas desenvolvidos, de forma detalhada, dando relevo às funcionalidades oferecidas por cada um desses comandos.

6.1 ESTRUTURA DE COMANDOS

Nesta API, todas as mensagens trocadas entre o PIC e o módulo BLE são codificadas segundo a norma ASCII, pelo que estas são processadas em ambos os extremos por um interpretador de comandos que reconhece esta codificação. A estrutura de comandos desenvolvida para a transmissão de mensagens consiste numa sequência de dados que seguem a base ilustrada na expressão 6.1.

$$\$C_p, Comm, SN, Action, \dots \# \quad (6.1)$$

Todos os comandos são iniciados por um carater especial, \$, e terminados pelo carater #. A escolha destes caracteres de iniciação e terminação devem-se à sua rara presença no tipo de dados existentes nas mensagens transmitidas, o que permite a deteção eficaz de comandos enviados, sem a necessidade de incluir um carater de escape. Associado a isto, estes comandos são também constituídos por 4 campos obrigatórios, sendo possível acrescentar parâmetros adicionais, se necessário. Os campos obrigatórios referidos correspondem a:

- Versão do processador de comandos (C_p): este campo determina qual o interpretador de comandos que deverá responder à mensagem enviada, permitindo que haja comandos semelhantes que podem ser interpretado de forma diferente por vários interpretadores diferentes. Este campo não é propagado para as camadas internas da API, sendo

retirado antes do processamento.

- Comando (Comm): este campo corresponde ao código do comando a ser executado pelo processador de comandos. Consoante o comando escolhido, o campo *action* pode ter diferentes significados.
- Número de sequência (SN): este campo corresponde ao número de sequência da mensagem. Este valor é sequencial e incrementado por ambos os terminais, por forma a permitir a deteção de mensagens perdidas e troca de mensagens de *acknowledgment*.
- Parâmetro de ação (Action): este campo corresponde ao primeiro parâmetro enviado ao processador de comandos, para um comando específico. Em determinados comandos, como será relevado posteriormente, este campo corresponde a um identificador de uma ação adicional, funcionando como um segundo comando, o que exige que sejam especificados parâmetros adicionais.

6.2 COMANDOS

A interface com a aplicação é dividida em dois subsistemas de comandos, descritos pela tabela 6.1, que consistem na adaptação do mecanismo de comandos previamente existente no sistema Bikeemotion, no caso do subsistema Bike, e no desenvolvimento de um novo sistema de comandos adicional, a que se deu o nome de subsistema extended, que permite implementar as novas funcionalidades descritas neste documento. Ao permitir a integração com o sistema de processamento de comandos previamente existente no *firmware* das bicicletas, permite que sejam migradas funcionalidades de acesso a informação já existente, com a vantagem de se recorrer agora a uma nova tecnologia de comunicação sem fios, BLE, enquanto que anteriormente apenas se poderia aceder a esta informação realizando a desmontagem parcial da bicicleta. Por outro lado, o subsistema extended permite a implementação de comandos de troca de mensagens de tamanho variável, dividindo em pequenos segmentos que conseguem ser transmitidos de uma só vez sobre a ligação BLE.

Nome	ID	Descrição
Bike	1	Gere os comandos que se referem às funcionalidades da bicicleta
Extended	2	Gere os comandos adicionais criados para o sistema

Tabela 6.1: Tabela dos processadores de comandos, respetivo ID na trama e descrição da sua função

6.3 SUBSISTEMA BIKE

O subsistema Bike corresponde à migração da estrutura de comandos previamente existente nas bicicletas, que se encontrava disponível através de uma ligação local a uma porta de

comunicações série UART, após uma desmontagem parcial da bicicleta. Com esta implementação, é possível a troca de comandos entre a aplicação móvel sobre uma ligação sem fios BLE e a bicicleta, sendo encaminhado internamente para o mesmo mecanismo de interpretação utilizado pela comunicação referida anteriormente. A tabela 6.2 apresenta todos os comandos atualmente utilizados, que serão descritos com mais detalhe nesta secção do documento.

Comando	Código	Descrição
Tampering	0	Informação do sensor de vandalismo
Acelerómetro	1	Informação das medições do acelerómetro
Temperatura	2	Informação das medidas do sensor de temperatura
Bateria	3	Informação do estado da bateria
Odómetro	4	Informação do sensor de velocidade nas rodas
LDR	5	Informação do detetor de luminosidade
Reset	6	Pedido de reinício da placa
Locks	7	Informação do estado das docas de carga
EEPROM	8	Pedido de operações na memória não volátil
Tech Screen	9	Pedido de entrada em modo técnico
Motor Control	10	Controlo do motor
USB	11	Informação sobre o USB
Torque	13	Informação sobre o sensor de torque
Assistência	14	Informação sobre o estado da assistência ao utilizador
Áudio	15	Informação sobre a interface áudio
GPS	16	Informação sobre o estado do GPS
Luzes	18	Informação sobre as luzes
Modem	19	Informação sobre o modem
RFID	21	Informação dos cartões lidos

Tabela 6.2: Tabela dos comandos existentes no subsistema Bike, respetivo ID na trama e descrição da sua função

6.3.1 Tampering

O comando de *tampering* destina-se a fazer a interação do sistema com o sensor de vandalismo. Como observado na expressão 6.2, este comando não aceita nenhum parâmetro adicional, sendo que o campo *Action* serve como seletor para a função pretendida.

$$\$1, 0, SN, Action\# \quad (6.2)$$

Atualmente, este comando aceita como funções disponíveis apenas o pedido de leitura do estado do sensor, correspondente ao código 0, sendo que a resposta enviada é a exemplificada

pela expressão 6.3, em que *value* corresponde ao valor devolvido pelo sensor, que pode tomar o valores 1 ou 0, se o sensor tiver sido disparado, ou não, respetivamente.

$$\text{\$1,0,SN,0,value\#} \quad (6.3)$$

6.3.2 Acelerómetro

O comando de acelerómetro destina-se a fazer pedidos de informação acerca desse sensor. O pedido a ser executado é o exemplificado pela expressão 6.4, sendo o parâmetro *Action* destinado ao tipo de pedido a ser executado.

$$\text{\$1,1,SN,Action\#} \quad (6.4)$$

Atualmente, os tipos de pedido que podem ser executados correspondem a obter o ID do sensor, com o código 0, e a obter os valores atuais do sensor, com o código 1. A resposta obtida pela execução destes comandos corresponde às expressões 6.5 e 6.6, respetivamente, em que o parâmetro ID da primeira resposta corresponde à identificação do sensor, e os parâmetros X, Y e Z, na segunda resposta, correspondem aos valores do acelerómetro para cada um dos eixos, sem processamento.

$$\text{\$1,1,SN,0,ID\#} \quad (6.5)$$

$$\text{\$1,1,SN,0,X,Y,Z\#} \quad (6.6)$$

6.3.3 Temperatura

O comando de temperatura destina-se a pedidos de informação ao sensor de temperatura existente no sistema. O pedido a ser executado, exemplificado na expressão 6.7, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1,2,SN,Action\#} \quad (6.7)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura de temperatura, com o código 0, sendo que a resposta obtida é exemplificado pela expressão 6.8, em que o parâmetro *value* corresponde ao valor devolvido pelo sensor, em graus Celsius, arredondado à unidade.

$$\text{\$1,2,SN,0,value\#} \quad (6.8)$$

6.3.4 Bateria

O comando de bateria destina-se a fazer pedidos de informação e ação no módulo de controlo da carga da bateria. O pedido a ser efetuado, exemplificado pela expressão 6.9, aceita como único parâmetro o tipo de ação a executar.

$$\text{\$1,3,SN,Action\#} \quad (6.9)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Obter valor de tensão: sendo a resposta obtida exemplificada pela expressão 6.10, esta ação permite obter o valor da tensão atual da bateria, em Volt, com 3 casas decimais;
- 1 - Obter valor de corrente: sendo a resposta obtida exemplificada pela expressão 6.11, esta ação permite obter o valor da corrente atual a entrar ou a sair da bateria, em Ampere, com 2 casas decimais;
- 2 - Obter valor de percentagem de bateria: sendo a resposta obtida exemplificada pela expressão 6.12, esta ação permite obter o valor da carga da bateria, em percentagem;
- 3 - Ativar o modo de carga: esta ação, ainda que não devolva nenhuma resposta, permite ativar o modo de carga da bateria, forçando a que seja permitido o fluxo de energia do carregador para a mesma;

$$\text{\$1, 3, SN, 0, voltage\#} \quad (6.10)$$

$$\text{\$1, 3, SN, 0, current\#} \quad (6.11)$$

$$\text{\$1, 3, SN, 0, percentage\#} \quad (6.12)$$

Os parâmetros *voltage*, *current* e *response*, nas expressões 6.10, 6.11, 6.12, respetivamente, corresponde ao valor da tensão, corrente e percentagem de carga da bateria, devolvido em resposta ao comando de temperatura.

6.3.5 Odómetro

O comando de odómetro destina-se a pedidos de informação ao sensor de velocidade existente no sistema. O pedido a ser executado, exemplificado na expressão 6.13, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 4, SN, Action\#} \quad (6.13)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura do sensor de velocidade, com o código 0, sendo que a resposta obtida é exemplificada pela expressão 6.14, em que o parâmetro *value* corresponde ao valor de velocidade devolvido pelo sensor, em Km/h, com 1 casa decimal.

$$\text{\$1, 4, SN, 0, value\#} \quad (6.14)$$

6.3.6 LDR

O comando de LDR destina-se a pedidos de informação ao sensor de luminosidade existente no sistema. O pedido a ser executado, exemplificado na expressão 6.15, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 5, SN, Action\#} \quad (6.15)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura do sensor de velocidade, com o código 0, sendo que a resposta obtida é exemplificado pela expressão 6.16, em que o parâmetro *value* corresponde ao valor de luminosidade devolvido pelo sensor.

$$\text{\$1, 5, SN, 0, value\#} \quad (6.16)$$

6.3.7 Reset

O comando de Reset destina-se a pedidos de reinício do sistema. O pedido a ser executado, exemplificado na expressão 6.15, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 6, SN, Action\#} \quad (6.17)$$

Este comando apenas aceita como tipo de ação o pedido de reinício do sistema, com o código 0, para o qual não é enviada nenhuma resposta.

6.3.8 Locks

O comando de *locks* destina-se a fazer pedidos de informação e ação no módulo de bloqueio da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.18, aceita como único parâmetro o tipo de ação a executar.

$$\text{\$1, 7, SN, Action\#} \quad (6.18)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Obter o estado do bloqueio: sendo a resposta obtida exemplificada pela expressão 6.19, esta ação permite obter o estado geral do bloqueio da bicicleta;
- 1 - Obter o estado individual dos bloqueios: sendo a resposta obtida exemplificada pela expressão 6.20, esta ação permite obter o estado de cada um dos bloqueios da bicicleta;
- 2 - Ativar o desbloqueio da bicicleta: este comando permite a ativação dos solenóides que desbloqueiam a bicicleta, não enviando nenhuma resposta ao comando;

$$\text{\$1, 7, SN, 0, status\#} \quad (6.19)$$

$$\text{\$1, 7, SN, 0, lockI1, lockI2, lockSW1, lockSW2\#} \quad (6.20)$$

O parâmetro *status*, na expressão 6.19, corresponde ao estado geral dos bloqueios da bicicleta, enquanto que os parâmetros *lockI1*, *lockI2*, *lockSW1* e *lockSW2*, na expressão 6.20, correspondem ao estado da alimentação de cada um dos 2 solenóides existentes e o estado dos correspondentes interruptores de contato, respectivamente.

6.3.9 EEPROM

O comando de *EEPROM* destina-se a fazer pedidos de ação no módulo de memória da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.21, aceita como parâmetros o tipo de ação a executar e, no caso do comando de leitura de dados, aceita também o índice da memória e o número de bytes a ser lidos.

$$\text{\$1, 8, } SN, Action[, wordIndex, numBytes]\# \quad (6.21)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Escrever as configurações por defeito: não enviando nenhuma resposta, esta ação permite escrever na memória as configurações por defeito da bicicleta;
- 1 - Ler dados da memória: sendo a resposta obtida exemplificada pela expressão 6.22, esta ação permite ler da memória um conjunto de dados, que são enviados no parâmetro *data* da resposta;

$$\text{\$1, 8, } SN, 0, data\# \quad (6.22)$$

6.3.10 Tech Screen

O comando de *Tech Screen* destina-se a fazer pedidos entrada e saída no modo de menu técnico da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.23, aceita como único parâmetro o tipo de ação a executar.

$$\text{\$1, 9, } SN, Action\# \quad (6.23)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Entrar em modo técnico: esta ação permite ativar o modo técnico da bicicleta;
- 1 - Sair do modo técnico: esta ação permite sair do modo técnico e repor a bicicleta em estado seguro, no menu de utilizador;

Ambos os tipos de ação enviam como resposta a exemplificada na expressão 6.24.

$$\text{\$1, 9, } SN, 0\# \quad (6.24)$$

6.3.11 Motor Control

O comando de *Motor Control* destina-se a fazer pedidos de informação e ação no módulo de controlo do motor elétrico da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.25, aceita como único parâmetro o tipo de ação a executar.

$$\text{\$1, 10, } SN, Action\# \quad (6.25)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Enviar o controlador do motor para modo de execução: sendo a resposta obtida exemplificada pela expressão 6.26, esta ação permite mudar o estado do controlador do motor para o estado ativo;
- 1 - Enviar o controlador do motor para modo de poupança de energia: sendo a resposta obtida exemplificada pela expressão 6.26, esta ação permite mudar o estado do controlador do motor para o estado de poupança de energia;
- 2 - Obter a versão de *firmware* do controlador de motor: sendo a resposta obtida exemplificada pela expressão 6.27, este comando permite ler a versão de *firmware* do controlador do motor, sendo enviada numa sequência de caracteres, no parâmetro *version* da resposta;
- 3 - Ativar o motor elétrico: sendo a resposta obtida exemplificada pela expressão 6.26, este comando permite ativar o motor elétrico da bicicleta;

$$\text{\$1, 10, SN, 0\#} \quad (6.26)$$

$$\text{\$1, 10, SN, 0, version\#} \quad (6.27)$$

6.3.12 USB

O comando de USB destina-se a pedidos de eco de teste da comunicação de dados pelo USB. O pedido a ser executado, exemplificado na expressão 6.28, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 11, SN, Action\#} \quad (6.28)$$

Este comando aceita como tipos de pedido apenas o comando de obtenção de estado do USB, com o código 0, sendo que a resposta obtida é exemplificado pela expressão 6.29.

$$\text{\$1, 5, SN, 0, value\#} \quad (6.29)$$

6.3.13 Torque

O comando de torque destina-se a pedidos de informação ao sensor de torque existente no sistema. O pedido a ser executado, exemplificado na expressão 6.30, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 13, SN, Action\#} \quad (6.30)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura do sensor, com o código 0, sendo que a resposta obtida é exemplificado pela expressão 6.31, em que o parâmetro *value* corresponde ao valor de torque devolvido pelo sensor.

$$\text{\$1, 13, SN, 0, value\#} \quad (6.31)$$

6.3.14 Assistência

O comando de assistência destina-se a pedidos de informação do estado da assistência a atuar no sistema. O pedido a ser executado, exemplificado na expressão 6.32, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 14, } SN, Action, \dots \# \quad (6.32)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura do estado da assistência, com o código 0, sendo que a resposta obtida é exemplificado pela expressão 6.33, em que o parâmetro *value* corresponde ao valor do *duty-cycle* atual da assistência ao motor.

$$\text{\$1, 14, } SN, 0, value\# \quad (6.33)$$

6.3.15 Áudio

O comando de *áudio* destina-se a fazer pedidos de informação e ação no módulo de áudio da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.34, aceita como único parâmetro o tipo de ação a executar.

$$\text{\$1, 15, } SN, Action\# \quad (6.34)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Executar a reprodução do som da buzina: sendo a resposta obtida exemplificada pela expressão 6.35, esta ação permite reproduzir o áudio da buzina;
- 1 - Obter o número de reproduções: sendo a resposta obtida exemplificada pela expressão 6.36, esta ação permite obter o número de reproduções do áudio da buzina;

$$\text{\$1, 15, } SN, 0\# \quad (6.35)$$

$$\text{\$1, 15, } SN, 0, count\# \quad (6.36)$$

Na expressão 6.36, o parâmetro *count* corresponde ao número de reproduções do áudio da buzina.

6.3.16 GPS

O comando de GPS destina-se a pedidos de informação ao sensor de GPS existente no sistema. O pedido a ser executado, exemplificado na expressão 6.37, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 16, } SN, Action\# \quad (6.37)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura do estado da localização por GPS, com o código 0, sendo que a resposta obtida é exemplificado

pela expressão 6.38, em que os parâmetros *alive*, *SU* e *SV* correspondem ao estado localização fixa, número de satélites em uso e número de satélites em vista, respetivamente.

$$\text{\$1, 16, SN, 0, alive, SU, SV\#} \quad (6.38)$$

6.3.17 Luzes

O comando de *Modem* destina-se a fazer pedidos de ação no módulo de iluminação da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.39, aceita como parâmetro o tipo de ação a executar.

$$\text{\$1, 18, SN, Action\#} \quad (6.39)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Ativar a iluminação frontal: esta ação permite ativar a iluminação frontal da bicicleta;
- 1 - Ativar a iluminação traseira: esta ação permite ativar a iluminação traseira da bicicleta;

Ambas os tipos de ação envia como resposta a exemplificada pela expressão 6.40.

$$\text{\$1, 18, SN, 0\#} \quad (6.40)$$

6.3.18 Modem

O comando de *Modem* destina-se a fazer pedidos de informação e ação no módulo de comunicações da bicicleta. O pedido a ser efetuado, exemplificado pela expressão 6.41, aceita como parâmetro o tipo de ação a executar e, no caso do tipo de ação ser o envio de um comando, aceita também o comando a ser enviado.

$$\text{\$1, 19, SN, Action[, command]\#} \quad (6.41)$$

Os tipos de ação permitidos por este comando correspondem a:

- 0 - Obter o estado da ligação de dados: sendo a resposta obtida exemplificada pela expressão 6.42, esta ação permite obter o estado da ligação 3G, ou GSM, da bicicleta;
- 1 - Enviar um comando para o modem: sendo a resposta obtida exemplificada pela expressão 6.43, esta ação permite enviar um comando para o modem, utilizando o campo adicional *command*;

$$\text{\$1, 19, SN, 0, status\#} \quad (6.42)$$

$$\text{\$1, 19, SN, 0\#} \quad (6.43)$$

6.3.19 RFID

O comando de RFID destina-se a pedidos de informação sobre os dados do leitor RFID existente no sistema. O pedido a ser executado, exemplificado na expressão 6.44, aceita como parâmetro o tipo de pedido a ser executado.

$$\text{\$1, 21, } SN, \text{ Action\#} \quad (6.44)$$

Atualmente, este comando aceita como tipos de pedido apenas o comando de leitura do número de cartões lidos pelo sistema, com o código 0, sendo que a resposta obtida é exemplificado pela expressão 6.45, em que o parâmetro *counter* corresponde ao valor da contagem de cartões lido pelo sistema.

$$\text{\$1, 21, } SN, 0, \text{ counter\#} \quad (6.45)$$

6.4 SUBSISTEMA EXTENDED

O subsistema extended corresponde ao conjunto de comandos desenvolvidos para adicionar as funcionalidades que se foram revelando necessárias para permitir toda a implementação dos requisitos. Para agrupar e processar estes novos comandos, decidiu-se desenvolver um novo processador de comandos, que permitisse a execução simultânea com o processador de comandos anterior, e que fosse retro-compatível com a estrutura dos comandos utilizados atualmente. Dessa forma, foram implementados os comandos descritos na tabela 6.3, sendo nesta secção abordados com um maior detalhe cada um deles.

Comando	Código	Descrição
Versão	0	Pedidos de informação sobre a versão do firmware
Bike Info	1	Pedidos de informação de dados da bicicleta
Message Handler	2	Gestão de mensagens
Action	3	Execução de atividades na bicicleta
Restart	4	Pedido de reinício
Modo técnico	5	Pedido e controlo do ecrã técnico

Tabela 6.3: Tabela dos comandos existentes no subsistema extended, respetivo ID na trama e descrição da sua função

6.4.1 Versão

O comando de versão permite à aplicação obter o a versão do *firmware* atualmente a correr no microcontrolador. Atualmente, este comando pode ser enviado com o campo *Action* com qualquer valor, uma vez que este não será utilizado pelo processador de comandos, podendo ser omitido da trama enviada, como pode ser observado na expressão 6.46.

$$\text{\$2, 0, } SN\# \quad (6.46)$$

A resposta enviada pelo microcontrolador corresponde ao descrito na expressão 6.47, em que os parâmetros X , Y e Z contêm os valores das versões *Major*, *Minor* e *Revision*, respetivamente.

$$\text{\$}2, 0, 0, X.Y.Z\# \quad (6.47)$$

6.4.2 Bike Info

O comando de *Bike Info* permite à aplicação efetuar um diagnóstico de sistema a todo o ambiente, pedindo informação sobre todos os sensores da bicicleta, dados do último cartão RFID lido, informações sobre a ligação ao servidor e dados de configuração.

Como é observável pela expressão 6.48, o parâmetro *Action* é facultativo, podendo ter qualquer valor.

$$\text{\$}2, 1, SN, [Action]\# \quad (6.48)$$

Após um pedido de informação ao sistema, este responde com a expressão 6.49, em que o parâmetro *size* corresponde ao número total de pacotes que o gestor de mensagens tem para enviar.

$$\text{\$}2, 1, 0, size\# \quad (6.49)$$

6.4.3 Message Handler

O comando de *Message Handler* permite a gestão da troca de mensagens entre a aplicação e o gestor de mensagens. Este comando permite aceder a segmentos da informação de forma indexada, permitindo o pedido de retransmissão de um pacote, caso não tenha sido recebido, como é representado pelo pedido na expressão 6.50, em que o parâmetro *num* representa o índice da mensagem que se pretende receber.

$$\text{\$}2, 2, SN, num\# \quad (6.50)$$

Após o pedido anteriormente referido, o sistema envia como resposta a expressão 6.51, em que o parâmetro *message* representa a mensagem que é enviada, no formato ASCII.

$$\text{\$}2, 2, 0, message\# \quad (6.51)$$

6.4.4 Action

O comando *Action* permite pedir um bloqueio ou desbloqueio de uma bicicleta, sem a necessidade de interagir fisicamente com a mesma. Como se pode observar pela expressão 6.52, este comando aceita um parâmetro *Action*, que pode tomar dois valores:

- 0: desbloqueio da bicicleta;
- 1: bloqueio da bicicleta;

$$\text{\$}2, 3, SN, Action\# \quad (6.52)$$

Em resposta a este comando, é enviada uma mensagem semelhante à expressão 6.53, em que *status* representa o sucesso do pedido, tomando um de dois valores:

- 0: pedido com sucesso;
- 1: falha no pedido;

$$\text{\$}2, 3, 0, status\# \quad (6.53)$$

6.4.5 Restart

O comando de *restart* permite à aplicação o agendamento de um reinício, repondo todo o sistema para um estado seguro. A sequência utilizada para despoletar este comando é descrita pela expressão 6.54, em que se pode observar que este comando não aceita nenhum parâmetro adicional.

$$\text{\$}2, 4, SN\# \quad (6.54)$$

Este comando é respondido utilizando a expressão 6.55, em que *status* corresponde ao valor 1 se o comando tiver sido bem reconhecido.

$$\text{\$}2, 4, 0, status\# \quad (6.55)$$

6.4.6 Modo técnico

O comando de modo técnico permite a passagem de uma bicicleta para o seu menu de informações técnica e a mudança de página. A estrutura necessária para a ativação deste comando corresponde à apresentada na expressão 6.56, sendo o parâmetro *Action* facultativo, podendo não ser enviado.

$$\text{\$}2, 5, SN, [Action]\# \quad (6.56)$$

Caso a bicicleta já se encontre em modo técnico, a execução deste comando permite percorrer as páginas do mesmo, sendo que se efetua a mudança para o ecrã de utilizador após serem percorridas todas as páginas. Caso esta esteja no menu de utilizador, dá-se entrada no menu técnico.

Testes e Validação

Após a finalização do desenvolvimento das funcionalidades previstas nos requisitos do projeto, foi necessário realizar um conjunto de testes que permitisse concluir sobre a robustez da ligação que é possível obter e impacto do consumo energético provocado pela inserção no sistema de um novo elemento de comunicações por radiofrequência.

Este capítulo consiste na descrição desses testes, assim como a sua execução, segundo os procedimentos descritos para cada um deles, finalizando com uma análise sobre os resultados obtidos.

7.1 TESTES DE DESEMPENHO

Os testes de desempenho realizados foram divididos em três tipos: testes de consumo, de estabilidade de ligação e de funcionamento do gestor de mensagens. Os testes de consumo permite auferir qual o impacto do consumo provocado pelo módulo RN4020, os testes de estabilidade de ligação permitem perceber se a ligação se mantém estabelecida em condições adversas e o teste ao gestor de mensagens permite perceber se os pacotes perdidos são retransmitidos para completar a mensagem transmitida.

7.1.1 Consumo energético

Para avaliar a relevância da descarga da bateria derivado da existência do módulo de comunicações Bluetooth Low Energy, efetuou-se a medição dos consumos energéticos das duas placas que compõem o *hardware* do sistema Bikeemotion nas condições expressas na tabela 7.1, em que são apresentados os resultados nas condições de inexistência de alimentação do módulo, tomado como referência para as medidas subsequentes, e em cada um dos estados de atividade do módulo de comunicações BLE:

Descrição	Consumo (mA)	Diferença para a referência (mA)
BLE OFF (Referência)	117.69	0.00
BLE SLEEP	117.69	0.00
BLE IDLE	118.21	+0.52
BLE ADVERT	122.55	+4.86
BLE CONNECTED	124.13	+6.44
BLE TRANSMIT	132.61	+14.92

Tabela 7.1: Consumo do sistema com o estado da ligação BLE

7.1.2 Estabilidade da ligação

Para testar a estabilidade da ligação Bluetooth Low Energy entre a aplicação móvel e o *hardware* efetuou-se a seguinte experiência:

1. Estabeleceu-se uma ligação entre o dispositivo móvel e a bicicleta;
2. Enviou-se o comando de desbloqueio;
3. Aguardou-se 30 segundos;
4. Enviou-se o comando de bloqueio;
5. Aguardou-se 30 segundos;
6. Repetiu-se 4 vezes os passos 3 a 6, movimentando o telemóvel pela sala;
7. Pediu-se o envio das informações do ficheiro de diagnóstico;
8. Repetiu-se o passo 8;

Em cada um dos passos anteriores, verificou-se se a ligação ainda se encontrava estabelecida, não tendo existido nenhuma quebra em nenhum dos passos. Verificou-se ainda se o *hardware* executava a ação do comando enviado em cada um dos passos, não tendo havido nenhum comando que não tivesse sido executado.

No caso do envio ds dados de diagnóstico, verificou-se o número de pacotes que foram retransmitidos, sendo que apenas um dos pacotes teve que ser retransmitido na primeira transferência de dados, não havendo nenhuma retransmissão na segunda transferência.

7.1.3 Funcionamento com mensagens grandes

Para testar o comportamento da aplicação nas transferências de mensagens de grandes dimensões (acima de 100 bytes), efetuou-se a transferência da informação de um ficheiro de diagnóstico da bicicleta através da ligação Bluetooth Low Energy. O ficheiro gerado corresponde a uma codificação JSON, tendo um tamanho de 1204 bytes.

O ficheiro transmitido foi dividido em 13 pacotes de dados de 100 bytes. Apesar do limite de dados num pacote BLE ser de 20 bytes, o módulo RN4020 tem um *buffer* interno de 128 bytes, pelo que é possível dimensionar os pacotes como sendo de 100 bytes em vez de 20 bytes.

A transmissão anterior foi realizada em duas condições diferentes, sendo a primeira num ambiente em que não existia uma elevada ocupação do espetro a 2.4 GHz e a segunda num

ambiente de testes em existia a presença de sinais provenientes de uma rede sem fios IEEE 802.11.

Observou-se que, na primeira situação, nenhum pacote foi retransmitido durante os testes, não tendo sido detetado nenhum pacote danificado, e que, na segunda situação, apenas um dos pacotes foi retransmitido.

7.1.4 Alcance

Para testar o comportamento da ligação à medida que a distância varia, efetuou-se dois testes: o primeiro consistiu na observação do valor do RSSI relatado pela aplicação no intervalo de 10 centímetros a 1 metro, em intervalos de 10 centímetros, de 1 a 10 metros, em intervalos de 1 metro e de 10 a 20 metros, com intervalos de 10 metros. Os resultados são apresentados na tabela 7.2 e avaliados com a ajuda da figura 7.1, que corresponde ao gráfico do RSSI em função da distância.

Distância (m)	RSSI (dBm)	Distância (m)	RSSI (dBm)
0.1	-38	2	-63
0.2	-40	3	-72
0.3	-46	4	-78
0.4	-44	5	-80
0.5	-50	6	-74
0.6	-51	7	-80
0.7	-48	8	-81
0.8	-60	9	-83
0.9	-62	10	-80
1	-64	20	-87

Tabela 7.2: Dados obtidos do RSSI para cada valor de distância

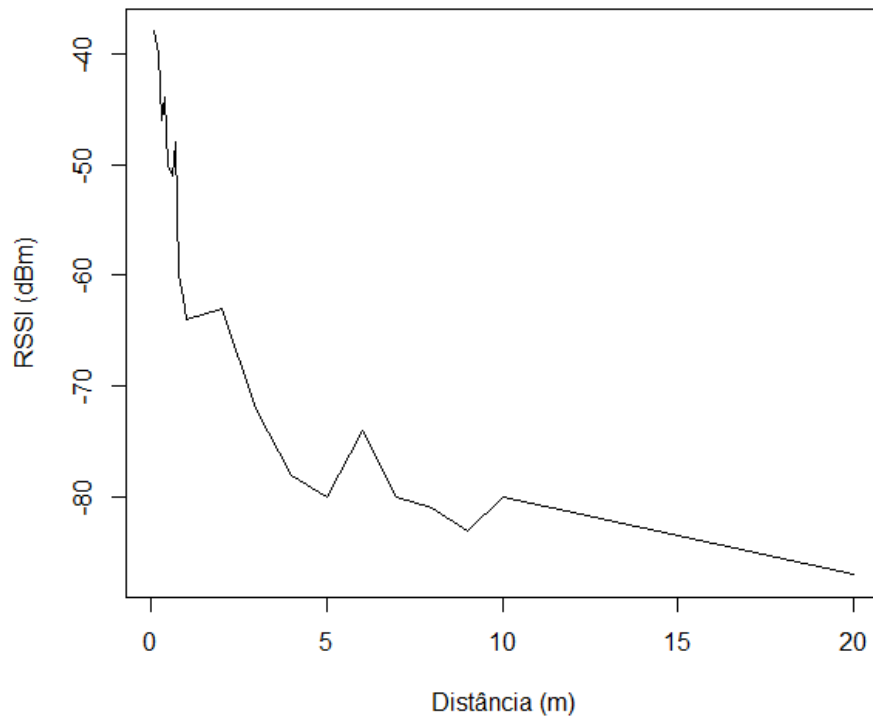


Figura 7.1: RSSI em função da distância

Não foram efetuadas medições acima da marca dos 20 metros, derivado da quebra de sinal observada no momento da medição da marca dos 30 metros.

7.1.5 Largura de Banda

Para testar a largura de banda do sistema, implementou-se no *hardware* um comando que envia uma mensagem de 100 bytes sempre que é recebido. A largura de banda é medida fazendo a troca de dados para 100 mensagens, medindo-se o tempo que demorou a fazer cada transferência. Os dados obtidos foram representados num gráfico, apresentado na figura 7.2.

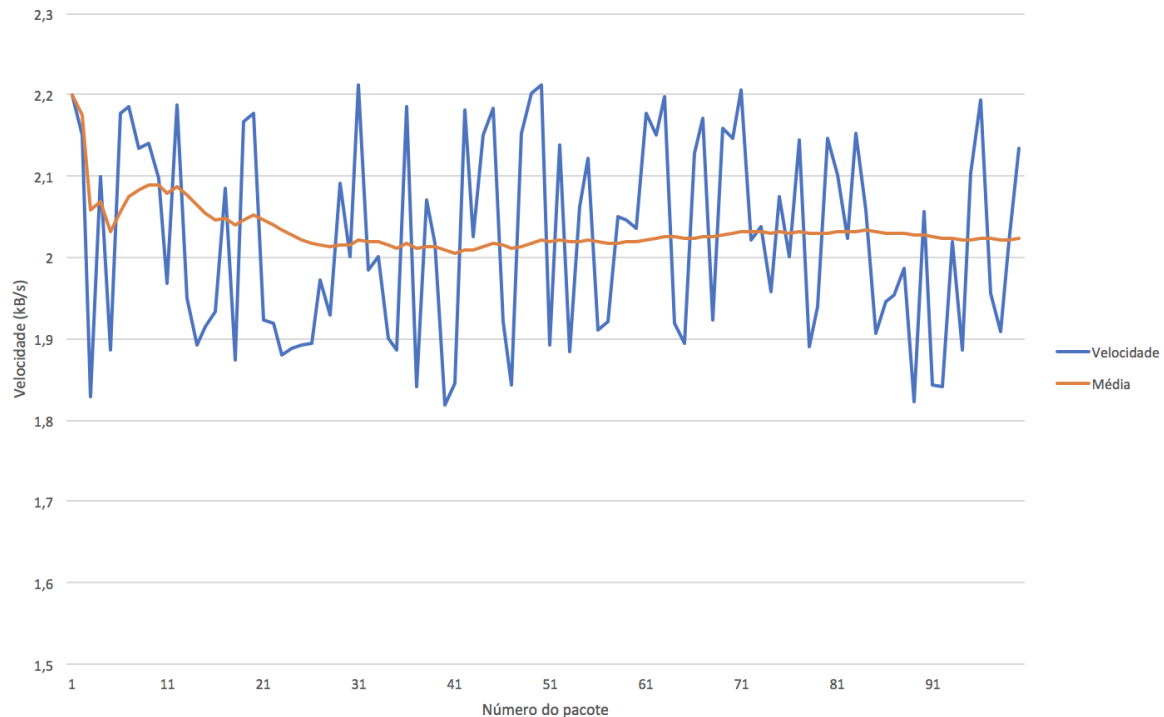


Figura 7.2: Largura de banda do sistema

Os valores observados no gráfico representam a largura de banda calculada, apresentando-se ainda um gráfico da média acumulada ao longo das amostras calculadas, representando a evolução do valor médio da taxa de transferência atingida.

7.2 ANÁLISE DOS RESULTADOS

Os testes efetuados demonstram que as funcionalidades desenvolvidas cumprem os requisitos, na medida em que a ligação Bluetooth Low Energy conseguida é suficientemente robusta para aguentar os ambientes de ruído simulados, não tendo demonstrado dificuldades em conseguir manter a ligação ativa mesmo quando os níveis de ruído foram aumentados para níveis pouco habituais.

Por outro lado, o gestor de mensagens funcionou como previsto, tendo dividido as mensagens em pacotes mais pequenos que permitissem o envio sem erros entre a aplicação móvel e o *hardware* abstraindo o utilizador das limitações da interface, tendo efetuado o reenvio do pacote que foi perdido durante a transmissão.

No que se refere à largura de banda conseguida com o sistema, esta ficou-se por um valor médio próximo dos 2 kB/s, bastante abaixo do máximo teórico de 7 kB/s do fabricante. Este resultado posse ser explicado pelos atrasos no processamento das mensagens, aliado à não otimização da velocidade de comunicação, uma vez que a solução implementava não necessita de uma elevada largura de banda. Um outro fator que influencia estes valores consiste nas variações das condições do meio, no que se refere a interferências externas e qualidade do sinal na comunicação.

Em relação ao alcance conseguido com um canal de comunicações Bluetooth Low Energy, foi possível apurar que a ligação é estável num raio de 20 metros, não sendo necessário estar ao lado do equipamento a que se pretende ligar.

Finalmente, os dados obtidos dos consumos energéticos adicionais impostos pela integração do módulo no *hardware* demonstram que, no estado de *sleep* do mesmo, o consumo adicional não é mensurável na escala de medição, sendo que o *data sheet* apresenta como consumo teórico uma corrente de 900 nA a 3.3V. Por outro lado, no estado de *IDLE*, o consumo atingiu a marca de 0.52 mA, o que, se assumirmos um intervalo de 4 horas entre duas cargas, implicaria uma descarga da bateria correspondente a 2.08 mAh, valor que é mínimo quando comparado com as capacidades da bateria que acompanha o sistema. Os restantes estados do módulo apenas acontecem em situações excecionais e por curtos espaços de tempo. Se for considerada uma intervenção realizada durante 2 horas por um técnico e um consumo adicional de corrente correspondente ao valor máximo de 14.92 mA, é possível verificar que o esforço de capacidade corresponde a 29.84 mAh, não provocando valores significativos de descarga da bateria.

Considerações Finais

8.1 CONCLUSÕES

O principal objetivo deste trabalho consistiu em implementar um mecanismo de comunicações sem fios de curto alcance, Bluetooth Low Energy, ao nível de hardware, firmware e aplicação móvel, no sistema da Bikeemotion previamente existente.

Esta implementação demonstra o nível de potencialidades que é possível obter com este novo canal de comunicações, principalmente por funcionar como um auxílio às interfaces previamente existentes.

Com o desenvolvimento do protocolo de comunicação, conseguiu-se demonstrar que é possível obter o mesmo nível de integração com o sistema utilizando a comunicação Bluetooth Low Energy que se poderia obter utilizando a ligação USB ou UART, dando origem a oportunidades de desenvolvimento de uma ferramenta completa de diagnóstico sem fios, que poderia ser implementada em equipamentos móveis genéricos e que permitiria à equipa técnica uma redução do tempo despendido.

8.2 TRABALHO FUTURO

Apesar de todo o desenvolvimento realizado neste trabalho, é necessário potenciar o uso da tecnologia que foi integrada no sistema. Para isso, é necessário que, em trabalho futuro, sejam desenvolvidas novas funcionalidades, ou que sejam migradas algumas que se encontram já implementadas em mecanismos alternativos e que não foram abordadas neste documento, como a atualização de firmware da placa mãe, que atualmente só pode ser feita através de uma ligação de dados GSM ou 3G.

Para tornar estas novas funcionalidades mais apelativas ao utilizador final, deve ser aperfeiçoada a interface gráfica da aplicação desenvolvida, permitindo novas funcionalidades como filtros para escolha das bicicletas ou o estabelecimento de grupos para diferenciar certas bicicletas.

Por outro lado, uma das funcionalidades que seria interessante explorar no futuro seria a de permitir que as bicicletas efetuassem diagnósticos automáticos e enviassem, em broadcast,

na forma de beacons, o seu estado atual, o que agilizaria bastante a detecção de bicicletas a necessitar de apoio técnico.

Uma vez que a ligação Bluetooth Low Energy pode sofrer interferências de outros elementos a trabalhar na mesma gama de frequências, seria valioso aproveitar o mecanismo de indexação das mensagens existente no gestor de mensagens e criar um mecanismo que permita o envio de mais que uma mensagem enquanto aguarda pelas confirmações, sendo que atualmente este envio está a ser realizado sequencialmente, sendo enviada uma mensagem apenas depois da receção de uma confirmação ou se tiver passado um intervalo de tempo sem ter chegado a confirmação, sendo a mensagem anterior retransmitida.

A fim de permitir um diagnóstico mais controlado dos problemas das bicicletas, deve ser desenvolvido um mecanismo para teste isolado de todos os elementos da mesma, permitindo executar ações no sistema, como ativar o motor, ver os sinais dos sensores ou efetuar a leitura do cartão RFID, permitindo ao mesmo tempo obter informações semelhantes às existentes no menu técnico das bicicletas.

Referências

- [1] B. Europe. (30 de abr. de 2017). EU Regulations for E-bikes & Pedelecs (Part 1), URL: <http://www.bike-eu.com/laws-regulations/artikel/2010/8/eu-regulations-for-e-bikes-pedelecs-part-1-1018836>.
- [2] U. Europeia. (30 de abr. de 2017). DIRECTIVA 2002/24/CE DO PARLAMENTO EUROPEU E DO CONSELHO, URL: <http://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=CELEX:32002L0024>.
- [3] E. B. Guide. (17 de fev. de 2017). Pedelec vs E-bikes, URL: <http://www.electric-bicycle-guide.com/pedelec.html>.
- [4] I. for Local Government. (jan. de 2011). In Focus: The Last Mile and Transit Ridership, URL: <http://www.ca-ilg.org/post/focus-last-mile-and-transit-ridership>.
- [5] C. Cities. (30 de abr. de 2017). European Best Practices in Bike Sharing Systems, URL: <http://www.cyclecities.eu/data-en/file/knowledge-resources/BSS%20v.1/European%20Best%20Practices%20in%20Bike%20Sharing%20Systems.pdf>.
- [6] PopVille. (25 de out. de 2017). Where had You Find A Dockless Bike Today?, URL: <https://www.popville.com/2017/10/whered-you-find-a-dockless-bike-today-vol-20/>.
- [7] S. Volcano. (25 de out. de 2017). Twitter Comment, URL: https://twitter.com/BE_DIZZLE/status/922502743817719808?s=09.
- [8] R. D. Deutschland, «Runde Sache», pp. 74–75, jun. de 2011.
- [9] NACTO. (31 de ago. de 2017). Number of Bike Share Systems in the US by Type, URL: [https://cdn.vox-cdn.com/thumbor/iPwnHz_cJj9TQFXEuCUcJ7_Y6X0=/800x0/filters:no_upscale\(\)/cdn.vox-cdn.com/uploads/chorus_asset/file/8123573/BikeShare_Type_1200_Feb21.png](https://cdn.vox-cdn.com/thumbor/iPwnHz_cJj9TQFXEuCUcJ7_Y6X0=/800x0/filters:no_upscale()/cdn.vox-cdn.com/uploads/chorus_asset/file/8123573/BikeShare_Type_1200_Feb21.png).
- [10] —, (31 de ago. de 2017). Bike Share in the US: 2010-2016, URL: <http://nacto.org/bike-share-statistics-2016/>.
- [11] —, (31 de ago. de 2017). Bike Share Growth in the US, URL: [https://cdn.vox-cdn.com/thumbor/EWYE-K8v4kZBR7f0G06IqppqJ28=/800x0/filters:no_upscale\(\)/cdn.vox-cdn.com/uploads/chorus_asset/file/8123403/BikeShareGrowth_1200_b.png](https://cdn.vox-cdn.com/thumbor/EWYE-K8v4kZBR7f0G06IqppqJ28=/800x0/filters:no_upscale()/cdn.vox-cdn.com/uploads/chorus_asset/file/8123403/BikeShareGrowth_1200_b.png).
- [12] —, (31 de ago. de 2017). Bike Share in the US 2010-2016, URL: [https://cdn.vox-cdn.com/thumbor/yW_yppF_S5lxJFfCf6fnETDZQ_4=/800x0/filters:no_upscale\(\)/cdn.vox-cdn.com/uploads/chorus_asset/file/8116131/Map_1200_Feb21_v2.png](https://cdn.vox-cdn.com/thumbor/yW_yppF_S5lxJFfCf6fnETDZQ_4=/800x0/filters:no_upscale()/cdn.vox-cdn.com/uploads/chorus_asset/file/8116131/Map_1200_Feb21_v2.png).
- [13] Bewegen. (1 de mai. de 2017). About Bewegen, URL: <http://bewegen.com/about-us/#our-partners>.
- [14] —, (1 de set. de 2017). Our Cities, URL: <http://bewegen.com/our-cities/>.
- [15] —, (20 de fev. de 2017). Our System, URL: <http://bewegen.com/our-system/>.
- [16] Wikimedia. (31 de ago. de 2017). Bixi St. Laurent, URL: https://upload.wikimedia.org/wikipedia/commons/a/ae/Bixi_St-Laurent.jpg.
- [17] dolcecity.com. (19 de fev. de 2017). BiciMAD: Guía para no perderse con las bicis de alquiler de Madrid, URL: <http://www.dolcecity.com/madrid/2014/07/bicimad-guia-para-no-perderse-con-las-bicis-de-alquiler.asp>.

- [18] C. D. Co. (19 de fev. de 2017). The Bike Share Bicycle Copenhagen ALMOST Had, URL: <http://www.copenhagenize.com/2013/05/the-bike-share-bicycle-copenhagen.html>.
- [19] ESB. (19 de fev. de 2017). e-Bike Connection Solution Systems, URL: <http://www.esb.bike/>.
- [20] Velobility. (19 de fev. de 2017). Bike Sharing Solutions, URL: <http://www.velobility.net/>.
- [21] —, (27 de jul. de 2017). VELOSPOT BIKES FOR THE CITY OF BIEL, URL: http://www.velobility.net/portfolio_page/velospot-bikes-for-the-city-of-biel/.
- [22] —, (27 de jul. de 2017). E-BIKES FOR THE UNIVERSITY ETH IN ZURICH, URL: http://www.velobility.net/portfolio_page/e-bikes-for-the-university-eth-in-zu%cc%88rich/.
- [23] —, (27 de jul. de 2017). GENEVEROULE, GENEVA, URL: http://www.velobility.net/portfolio_page/geneveroule-geneva/.
- [24] —, (27 de jul. de 2017). LAUSANNEROULE, LAUSANNE, URL: http://www.velobility.net/portfolio_page/lausanneroule-lausanne/.
- [25] —, (27 de jul. de 2017). RENTAL BICYCLES FOR THE CITY OF KASSEL, URL: http://www.velobility.net/portfolio_page/rental-bicycles-for-the-city-of-kassel/.
- [26] —, (27 de jul. de 2017). RENTAL BICYCLES FOR THE CITY OF MAINZ, URL: http://www.velobility.net/portfolio_page/rental-bicycles-for-the-city-of-mainz/.
- [27] —, (27 de jul. de 2017). RENTAL BIKES FOR THE CITY OF BERN AND THUN, URL: http://www.velobility.net/portfolio_page/rental-bikes-for-the-city-of-bern-and-thun/.
- [28] —, (27 de jul. de 2017). SERVICE BIKES FOR THE CITY OF WINTERTHUR, URL: http://www.velobility.net/portfolio_page/bikes-for-the-city-of-winterthur/.
- [29] P. Portal. (15 de set. de 2017). Lidl-Bike startet am 5. März in Berlin, URL: <http://www.presseportal.de/pm/58227/3568755>.
- [30] P. Kinney. (30 de abr. de 2017). Wireless Control that Simply Works, URL: http://web.archive.org/web/20100411233226/http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=5162.
- [31] Fraunhofer. (30 de abr. de 2017). Wireless data at top speed, URL: <https://www.fraunhofer.de/en/press/research-news/2012/october/wireless-data-at-top-speed.html>.
- [32] B. SIG. (1 de set. de 2017). Core Specifications, URL: <https://www.bluetooth.com/specifications/bluetooth-core-specification>.
- [33] E. Design. (1 de set. de 2017). Design With Bluetooth For The Sports and Fitness Market, URL: <http://www.electronicdesign.com/communications/design-bluetooth-sports-fitness-market>.
- [34] Argenox. (1 de set. de 2017). A BLE Advertising Primer, URL: <http://www.argenox.com/a-ble-advertising-primer/>.
- [35] MbitLab. (1 de set. de 2017). BLE Link Layer State Machine, URL: <http://community.mbitlab.com/discussion/1668/ble-link-layer-state-machine>.
- [36] Firefly. (1 de set. de 2017). FireBLE, URL: http://wiki.t-firefly.com/index.php/FireBLE/First_app_for_IOS/en.
- [37] Microchip. (5 de out. de 2017). Bluetooth® Low Energy Module, URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/50002279B.pdf>.
- [38] —, (5 de out. de 2017). RN4020 Bluetooth Low Energy Module User's Guide, URL: ww1.microchip.com/downloads/en/DeviceDoc/70005191B.pdf.
- [39] —, (22 de nov. de 2017). PIC32MZ1024ECM144, URL: <http://www.microchip.com/wwwproducts/en/PIC32MZ1024ECM144>.
- [40] —, (5 de out. de 2017). PIC32MZ Embedded Connectivity (EC) Family Silicon Errata and Data Sheet Clarification, URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/80000588J.pdf>.

- [41] —, (22 de nov. de 2017). PIC32MZ1024EFM144, URL: <http://www.microchip.com/wwwproducts/en/PIC32MZ1024EFM144>.
- [42] A. D. Blog. (1 de set. de 2017). Announcing the Android 1.0 SDK, release 1, URL: <https://android-developers.googleblog.com/2008/09/announcing-android-10-sdk-release-1.html>.
- [43] Android. (31 de ago. de 2017). Android stack Governance Philosophy, URL: http://3.bp.blogspot.com/-iA9YEjrKmc/V09OROp9bLI/AAAAAAAAUY/C7jYRnwf3NM/s1600/androidsexample-android_framework_details.png.
- [44] —, (1 de set. de 2017). Bluetooth Low Energy, URL: <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>.
- [45] IDC. (1 de set. de 2017). Smartphone OS Market Share, 2017 Q1, URL: <http://www.idc.com/promo/smartphone-market-share/os>.
- [46] Javatpoint. (31 de ago. de 2017). Android Activity Lifecycle, URL: <https://www.javatpoint.com/images/androidimages/Android-Activity-Lifecycle.png>.

